ARL-TR-9648 ● Feb 2023

Dialogue-AMR Annotation Guidelines

by Claire Bonial, Taylor Hudson, Lucia Donatelli, David Traum, Clare Voss, Mitchell Abrams, and Austin Blodgett

NOTICES

Disclaimers

ARL-TR-9648 ● Feb 2023

Dialogue-AMR Annotation Guidelines

Claire Bonial and Clare Voss
DEVCOM Army Research Laboratory

Taylor Hudson
Oak Ridge Associated Universities

Lucia Donatelli
Saarland University

David Traum
Institute for Creative Technologies,
University of Southern California

Mitchell Abrams
Tufts University

Austin Blodgett
Institute for Human & Machine Cognition

REPORT DOCUMENTATION PAGE						*Form Approved OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*
February 2023

2. REPORT TYPE
Technical Report

3. DATES COVERED (From - To)
May 2018–September 2022

4. TITLE AND SUBTITLE
Dialogue-AMR Annotation Guidelines

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Claire Bonial, Taylor Hudson, Lucia Donatelli, David Traum, Clare Voss, Mitchell Abrams, and Austin Blodgett

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
DEVCOM Army Research Laboratory
ATTN: FCDD- RLA-IC
Adelphi, MD 20783-1138

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-9648

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
ORCID ID: Claire Bonial, 0000-0002-3154-2852

14. ABSTRACT
This report describes a schema that enriches Abstract Meaning Representation (AMR) in order to provide a semantic representation for facilitating Natural Language Understanding in dialogue systems. We explore dialogue in several domains of human–robot interaction, where a conversational robot is engaged in collaborative tasks with a human partner. AMR offers a valuable level of abstraction of the propositional content of an utterance; however, it does not capture the illocutionary force or speaker's intended contribution in the broader dialogue context (e.g., make a request or ask a question), nor does it capture tense or aspect. To address these limitations, we develop an inventory of speech acts suitable for our domain, and present "Dialogue-AMR," an enhanced AMR that represents not only the content of an utterance, but the illocutionary force behind it, as well as tense and aspect.

15. SUBJECT TERMS
Natural Language Processing, human–robot interaction, dialogue, dialogue systems, meaning representation, Military Information Sciences

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					Claire Bonial
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (Include area code)
Unclassified	Unclassified	Unclassified	UU	76	301-394-1431

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

1. Introduction

This report describes a schema that enriches Abstract Meaning Representation (AMR) in order to provide a semantic representation for facilitating Natural Language Understanding (NLU) in dialogue systems. AMR offers a valuable level of abstraction of the propositional content of an utterance; however, it does not capture the *illocutionary force*, or in other words, the speaker's intended contribution in the broader dialogue context (e.g., to make a request or ask a question), nor does AMR capture *tense* or *aspect*, which tell us about the time and completion status of an action. We explore dialogue in several domains of collaborative dialogue and specifically human–robot interaction, where a conversational robot is engaged in physically situated tasks (e.g., reconnaissance or manipulating objects) with a human partner. To address the limitations of AMR, which we will refer to as "Standard-AMR" in this report, we develop an inventory of *speech acts* as well as a representation of *tense*[*] and *aspect* information, culminating in "Dialogue-AMR." This results in an enhanced AMR that represents not only the propositional content of an utterance, but the illocutionary force behind it, as well as the time and completion status of the primary actions. We use both manual and automatic methods to construct the "DialAMR" corpus—a corpus with dialogue data from three domains:

1) Human–robot collaborative dialogue for search and navigation

2) Human–human collaborative dialogue for building blocks structures in the virtual Minecraft gaming environment

3) Human–robot collaborative dialogue for locating and manipulating objects

In the DialAMR corpus, all utterances are annotated with both the Standard-AMR and our enriched Dialogue-AMR schema.

The contributions of the present research are the following:

- A set of speech acts finalized and situated in a taxonomy
- The refinement of the Dialogue-AMR annotation schema to provide coverage of novel language
- The creation of the DialAMR corpus, a collection of dialogues to which the new Dialogue-AMR schema has been applied, which can be used for

[*] Our annotation captures temporal information. We also use "tense" informally for this annotation.

training and evaluating automatic "parsers" or systems that can output the enhanced representation.

The DialAMR corpus constitutes one of the first corpora of dialogue annotated with AMR and allows for comparison of both (Standard and Dialogue) AMR schemas on the same data. Although some of the domain-specific extensions are tailored to our human–robot dialogue application, the addition of illocutionary force to AMR is useful for many applications of human-agent conversation. Furthermore, the general paradigm of extending AMR is useful for applications that underspecify some linguistic distinctions while retaining others.

Finally, we are leveraging DialAMR to develop an NLU pipeline (Fig. 1) that contains both a general-purpose representation language (Standard-AMR) as well as Dialogue-AMR, which is more amenable to inferences that a robot needs to make when engaged in a collaborative navigation task. This pipeline converts automatically generated Standard-AMR graphs of the input language into Dialogue-AMR graphs augmented with tense, aspect, and speech act information. The pipeline addresses a frequent dilemma in designing meaning representations for limited-domain dialogue systems—whether to preserve a general-purpose representation that is adequate for capturing most language expressions, or whether to focus on only the small subset that the system will be able to deal with. The former can make the representations more complex, the language interpretation more ambiguous, and the system-specific inference more difficult. The latter subset approach addresses these problems but may lose the ability to transfer to even a very similar domain and may require more in-domain data than is available for a new project. Our two-step pipeline maintains the advantages of each approach.

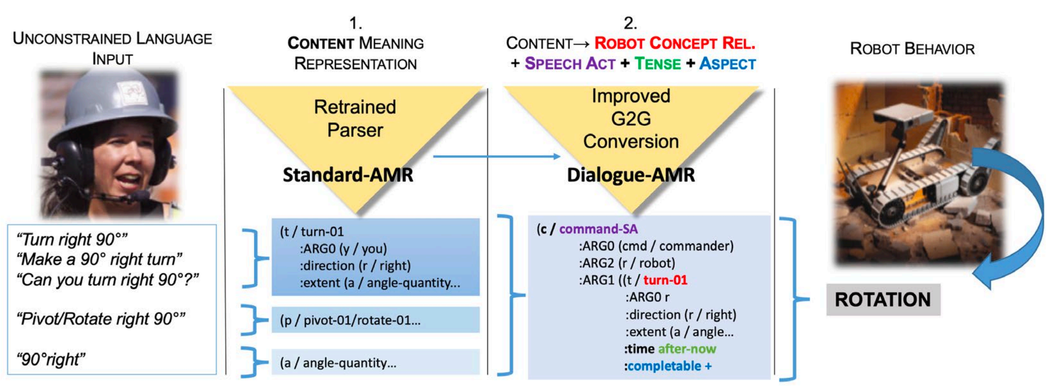

Fig. 1 In the two-step NLU pipeline, unconstrained natural language input (left) is first automatically parsed into Standard-AMR (1), capturing the content of the utterance, and then automatically converted into Dialogue-AMR (2), with the added elements of the speech act, tense, and aspect, and the conversion of various action predicates in the original input (as in 1) into a "robot concept relation" (as in 2, turn-01) that maps to one of the robot's executable behaviors (right) (here, ROTATION).

2. Background

Pragmatic information is critical in dialogue with a conversational agent. For example, a request for information and a request for action serve distinct dialogue functions. Similarly, a promise regarding a future action and an assertion about a past action will update the conversational context in very different ways. In our problem space, which involves a robot completing collaborative tasks with a human partner, the robot communicates about actions it can take in the environment, such as moving, searching, and reporting back. While the robot can be insensitive to many lexical differences, such as those between the movement terms *go*, *move*, and *drive*, it needs to understand specific parameters of the instructions, such as how far to go and when, as well as to communicate and discuss the status of a given task. Additionally, the robot needs to understand the illocutionary force of incoming communications that may be commands, suggestions, or clarifications.

To address these needs, we introduce a detailed and robust schema, building on Standard-AMR (Fig. 2a), for representing illocutionary force, tense, and aspect in our "Dialogue-AMR" (Fig. 2b). This expands and refines previous work that proposed basic modifications for how to annotate speech acts and temporal and aspect information within AMR (Bonial et al. 2019a, 2019b).

```
(a) (d / drive-01 :mode imperative
      :ARG0 (y / you)
      :destination (d2 / door))
(b) (c / command-SA
      :ARG0 (c2 / commander)
      :ARG2 (r / robot)
      :ARG1 (g / go-02 :completable +
        :ARG0 r
        :ARG3 (h / here)
        :ARG4 (d/ door)
        :time (a2 / after
            :op1 (n / now))))
```

Fig. 2 The utterance *Drive to the door* represented in (a) Standard-AMR form, (b) Dialogue-AMR form

In the sections that follow, we describe Standard-AMR and what motivates the design of Dialogue-AMR.

2.1 Standard-AMR

AMR is a formalism for sentence semantics that abstracts away from some surface syntactic idiosyncrasies (Banarescu et al. 2013); for example, whether a semantic concept is expressed in language as a single word (e.g., *offer*) or a more complex expression (e.g., *make an offer*), or whether another semantic concept is expressed as a noun (e.g., *my fear*) or an adjective (e.g., *I was afraid*). The Standard-AMR for each sentence is represented by a rooted directed acyclic graph in which the nodes identify entities, events, properties, and states. Leaf nodes in the graph are labeled with a variable and concept (e.g., (r / robot)). For relational concepts, in particular, the terms in Standard-AMR are drawn from a lexicon (shared with PropBank; Palmer et al. 2005) composed of numbered senses of a relation, each of which lists a set of numbered participant roles or "Args" (Arg0-Arg5). For ease of creation and manipulation, annotators work with Standard-AMR graphs written out in a formatted text notation from the PENMAN project (Penman Natural Language Group 1989), which is the notation used in this report (e.g., Fig. 2a). Since originally introduced, the AMR formalism for easily read and computationally tractable semantic structures has been used to support NLU, generation, and summarization (Liu et al. 2015; Pourdamghani et al. 2016), as well as machine translation (Langkilde and Knight 1998), question answering (Mitra and Baral 2016), information extraction (Pan et al. 2015), and biomedical text mining (Garg et al. 2016; Rao et al. 2017; Wang et al. 2017).

Standard-AMR provides an appropriate level of abstraction for NLU in our human–robot dialogue application. As the goal of semantic research in general is to capture core facets of meaning unrelated to surface structure variations of sentences; the same underlying Standard-AMR concept may be realized on the surface alternatively as a noun (*a left turn*), verb (*turn to the left*), or light verb construction (*make a left turn*). This mapping of various surface forms with the same meaning into the same Standard-AMR concept is well suited to our application: the robot has only a limited number of executable behaviors that it can perform, and so any user utterance must ultimately be mapped to a simple, yet structured representation with concept relations that the robot can execute. In turn, the robot only needs to communicate back to the user regarding those same concepts. Thus, Standard-AMR smooths away many syntactic and lexical features that are unimportant to the robot. Furthermore, we can leverage existing AMR *parsers*, that is, the software that automatically outputs the Standard-AMR of a sentence, to obtain an initial semantic interpretation of user utterances and bypass the need for a preliminary syntactic parse.

2.2 Gaps and Motivation for Dialogue-AMR

We began our assessment of Standard-AMR for human–robot dialogue by producing a small, randomly selected sample (137 sentences) of gold standard, manual annotations (provided by one senior and two recently trained AMR annotators), based on guidelines used in the production of the Linguistic Data Consortium (LDC) AMR releases.† We then examined how effectively these gold, guideline-based AMRs can capture the distinctions of interest for human–robot dialogue and how accurately two available AMR parsers can generate those gold annotations.

The human–robot dialogue was sampled from the Situated Corpus of Understanding Transactions (SCOUT), a US Army Combat Capabilities Development Command (DEVCOM) Army Research Laboratory–collected corpus of interactions between human commanders and a remotely located robot collaborating in reconnaissance tasks (Marge et al. 2016). Common instructions in the corpus include *Move forward 10 feet*, *Take a picture*, and *Turn right 45 degrees*. People also used landmark-based instructions such as *Move to face the yellow cone*, and *Go to the doorway to your right*, although these were less common than the metric-based instructions (Marge et al. 2017). In response to these instructions, common feedback from the robot to the human commanders would be indications that an instruction will be carried out (*I will move forward 10 feet*), is in progress (*Moving...*), or completed (*I moved forward 10 feet*).

The first problematic gap we noted in our assessment of Standard-AMR parses for human–robot dialogue is that neither their output nor the current AMR guidelines for annotating their training corpora make temporal/aspect distinctions. As a result, the three types of feedback from the robot, as just mentioned as common to SCOUT, are represented identically under the current guidelines (see Fig. 3). The distinctions between a promise to carry out an instruction in the future, a declarative statement that the instruction is currently being carried out, and an acknowledgment that it has been carried out, are critical for conveying the robot's current status in a live system; thus, it is crucial that these distinctions are represented clearly and not lost.

```
(m / move-01
      :ARG0 (i / i)
      :direction (f / forward)
      :extent (d / distance-quantity
            :quant 10
            :unit (f2 / foot)))
```

Fig. 3 Identical Standard-AMR for *I will move/I am moving/I moved forward...10 feet*

An additional gap that we noted relates to marking *imperatives*, in other words commands, which in English are characterized by a lack of an explicit grammatical

†Guidelines available for download here: https://github.com/amrisi/amr-guidelines/blob/master/amr.md; LDC release of AMR 3.0 (LDC2020T02) available here: https://catalog.ldc.upenn.edu/LDC2020T02

subject: *Go to the door*, as compared to the indicative sentence ***He** went to the door.* Although the imperative *Move forward 10 feet* should receive an AMR marker :MODE IMPERATIVE, our evaluation of the existing Standard-AMR parsers JAMR (Flanigan et al. 2014) and CAMR (Wang et al. 2015) showed that parser output does not include this marker as it is rare, if not entirely missing, from the AMR training corpora released by LDC. As a result, the command to *Move forward 10 feet* also received the identical AMR (Fig. 3) in parser output. While this indicates that additional training data is needed that includes imperatives, this also speaks to a larger issue of Standard-AMR: the existing representation is very limited with respect to speech act information such as marking commands. Current AMR guidelines include :MODE IMPERATIVE and represent questions through the presence of the constant AMR-UNKNOWN standing in for the concept or polarity being questioned. By default, all unmarked cases are assumed to be assertions. We found that more fine-grained speech act information is needed for human–robot dialogue so that intelligent agents can be trained to recognize when they are being commanded to do something, asked a question, or told some fact/observation. Note that these different types of input to the robot require different response behaviors.

3. Dialogue-AMR Annotation Elements

To develop an augmentation of Standard-AMR that addresses the requirements of human–robot dialogue, we iteratively refined an inventory of speech acts, capturing what a speaker is trying to do with their utterance (e.g., command, question) (Section 3.1), and introduced tense and aspect representations, capturing the time and completion status of instructed actions (Section 3.2). These additional elements of meaning are brought together in our annotation schema for Dialogue-AMR, in which the propositional semantic content is also normalized by replacing a variety of lexical items in the input language (e.g., *turn, pivot, rotate*) with an assigned relation (e.g., turn-01) that maps to a single robot concept corresponding to one of its executable behaviors (e.g., ROTATION) (Section 3.3).

3.1 Speech Act and Introduction of Speaker, Addressee

Linguistic theory often distinguishes between the "propositional content" of language, which can be thought of as what Standard-AMR largely represents, and the "illocutionary force" of language, which can be thought of as a separate level of meaning—what the speaker intends to do with their utterance in a conversational context. We aimed to represent both levels of meaning in Dialogue-AMR but needed to establish how to represent illocutionary force. To begin, we established an inventory of illocutionary forces, or "speech acts," observed in our data and noted in related work. We embraced much of the higher-level categorization and labeling of speech acts outlined in the seminal work of Searle (1969), including the basic categories of Assertions (termed "Representatives" by Searle), Commissives, Directives, and Expressives. Additionally, based on the Bunt et al. (2012) ISO standard, we introduced an early distinction in classifying our speech acts between Information-Transfer Functions and Action-Discussion Functions (see Fig. 4).

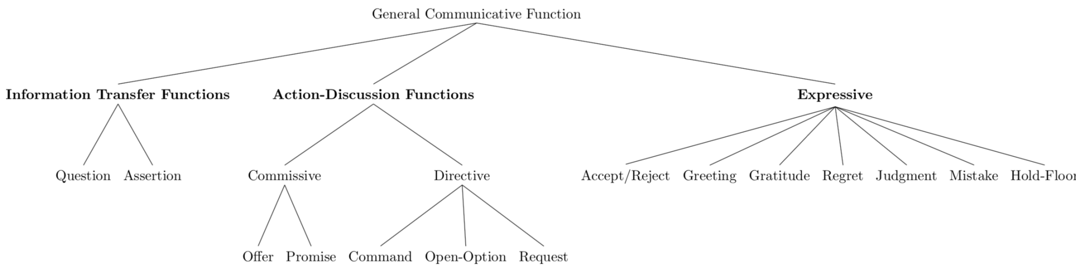

Fig. 4 Dialogue-AMR speech act taxonomy

In terms of dialogue function, this division allowed us to monitor the status of distinct dialogue contexts. For Information-Transfer types, we were able to monitor the quantity and quality of general-purpose information exchanged in the dialogue that is relevant to the larger task at hand. For example, *Robot, do you speak any foreign languages?* may not directly impact a current task, but it introduces information into the dialogue that may be useful at a later point. For Action-Discussion types, we could assess the status of individual tasks as the dialogue progressed. For example, *Moving to the wall* and *I moved to the wall* convey two points on a timeline related to current task completion. For Expressive types, we were able to model the changing relationship between interlocutors—for example, how utterances of gratitude, acceptance or rejection, and admission of mistakes impacted the level of trust between the two interlocutors.

Beyond these initial higher-level categories, we iteratively refined the speech act categories needed for our domain based upon rounds of surveying and annotating our data. These iterations began with the annotation of "dialogue moves" over participant instructions only (Marge et al. 2017) and evolved with varying numbers and types of speech acts (Bonial et al. 2019a, 2019b), finally culminating in the inventory set forth here.

In delineating and defining our speech acts, we focus on the effects of an utterance relating to Belief and Obligation (Traum 1999; Poesio and Traum 1998). These are not mutually exclusive, and utterances can and do often convey both the commitment to a Belief and evoke an Obligation in either the speaker or the hearer to do something or to respond in a particular way. We focus on these pragmatic effects as they are critical for agents navigating and searching an environment while also participating in a dialogue—in planning their dialogue moves, agents can choose to pursue either goals or obligations and must reason about these notions so that their choice can be explained. Mutual beliefs about the feasibility of actions and the intention of particular agents to perform parts of that action are captured in the notion of Committed, which is a social commitment to a state of affairs, rather than an individual commitment (Traum 1999). Definitions of our speech acts are given in Table 1.

Table 1 Dialogue-AMR speech act lexicon

Speech acts	Dialogue-AMR relations	Commitments and Obligations
Question	Question-SA	Speaker (S) committed to desire to know answer; Addressee (A) obliged to respond to question
Assertion	Assert-SA	S committed to a state of affairs
Offer	Offer-SA	S committed to feasibility of plan of action; A obliged to consider action and respond
Promise	Promise-SA	S committed to feasibility of plan of action and obliged to do action
Command	Command-SA	S committed to desire for A to do something and feasibility of action; A obliged to do action
Open-Option	Open-Option-SA	S committed to feasibility of action(s)
Request	Request-SA	S committed to desire for A to do something and feasibility of action; A is obliged to consider action and respond[8]
Accept/Reject	Accept-SA	S committed to a state of general acceptance or rejection[9]
Greeting	Greet-SA	S committed to recognizing presence of A and willingness to interact
Gratitude	Thank-SA	S committed to state of gratitude
Regret	Regret-SA	S committed to state of regret
Judgment	Judge-SA	S committed to evaluate stance
Mistake	Mistake-SA	S committed to acknowledging error
Hold Floor	Hold-Floor-SA	S committed to holding conversational floor for continued speech

Table 1 also lists the range of relations integrated into the Dialogue-AMR to represent the speech act. Unlike the numbered relations of Standard-AMR, we propose a new set of speech act relations all ending with -SA. Although we initially explored adopting existing AMR relations that best fit with each speech act (e.g., QUESTION-01, COMMAND-02) (Bonial et al. 2019b), we ultimately opted instead to introduce new relations marked –SA. In this way, the Dialogue-AMR clearly distinguishes what portion represents propositional content and what portion represents the illocutionary force. Additionally, we found that existing Standard-AMR relations were inconsistent in the argument structure representing the speaker, addressee, and content of the speech act. For example, while COMMAND-02 represents the addressee or impelled agent as ARG1 and the impelled action as ARG2, we found ASSERT-02 represents the addressee as ARG2 and the content of the assertion as ARG1. Our roles in our speech acts maintain the following consistent argument structure (as seen in Fig. 2b):

Arg0: Speaker

Arg1: Content

Arg2: Addressee

The roles of ARG0 and ARG2 correspond consistently, in all relations, to speaker and addressee, respectively; the semantics of the ARG1 shifts depending upon the particular speech act. For example, the ARG1-content of COMMAND-SA is an action, whereas the ARG1-content of REGRET-SA is the stimulus of the mental state, or the thing regretted.

3.2 Tense and Aspect Overview

There are patterned interactions between tense, aspect, and illocutionary force that are critical for conveying the robot's current status in our domains. These include the distinctions between a promise to carry out an instruction in the future, a declarative statement that the instruction is being carried out currently, and an acknowledgment that it was carried out in the past. Standard-AMR lacks information that specifies *when* an action occurs relative to speech time and whether or not this action is completed (if a past event) or able to be completed (if a future event). We integrate tense and aspect information into Dialogue-AMR by adopting the annotation schema of Donatelli et al. (2018), who propose a five-way division of temporal annotation and four multi-valued categories for aspectual annotation, that fits seamlessly into the propositional content in existing AMR annotation practice. We reduced the authors' proposed temporal categories to three in order to capture temporal relations before, during, and after the speech time. In addition to the aspectual categories proposed by Donatelli et al. (2018), we added the category :COMPLETABLE +/- to signal whether or not a hypothetical event has an end-goal that is executable for the robot (see Donatelli et al. 2019 for a sketch of this aspectual category). Our categories for temporal and aspectual annotation are listed in Fig. 5.

TEMPORAL ANNOTATION

```
:time
 1. (b / before
       :op1 (n / now))
 2. (n / now)
 3. (a / after
       :op1 (n / now))
```

ASPECTUAL ANNOTATION

```
:stable +/-
:ongoing +/-
:complete +/-
:habitual +/-
:completable +/-
```

Fig. 5 Three categories for temporal annotation and five categories for aspectual annotation are used to augment existing Standard-AMR for collaborative dialogue

Notably, this annotation schema is able to capture the distinctions missing in Fig. 3, as shown in Fig. 6. There the schema conveys distinct temporal information about a MOVEMENT event relative to the future, present, and past. It also communicates aspectual information related to the boundedness of the event (i.e., whether or not the event is a future event with a clear beginning and endpoint, a present event in progress toward an end goal, or a past event that has been completed from start to finish).

```
1. (m / move-01 :completable +
      :ARG0 (i / i)
      :direction (f / forward)
      :extent (d / distance-quantity
            :quant 10
            :unit (f2 / foot))
      :time (a / after
            :op1 (n / now)))

2. (m / move-01 :ongoing + :complete -
      :ARG0 (i / i)
      :direction (f / forward)
      :extent (d / distance-quantity
            :quant 10
            :unit (f2 / foot))
      :time (n / now))

3. (m / move-01 :ongoing - :complete +
      :ARG0 (i / i)
      :direction (f / forward)
      :extent (d / distance-quantity
            :quant 10
            :unit (f2 / foot))
      :time (b / before
            :op1 (n / now)))
```

Fig. 6 Updated AMRs for (1) *I will move...*, (2) *I am moving...*, and (3) *I moved....* New temporal and aspectual information is bold-faced.

There should be at least one tense and aspect relation annotation on each Dialogue-AMR instance. Annotation placement and other annotation details are provided in Section 4.

3.3 Robot Concept Relation

While the Standard-AMR abstracts away from some idiosyncratic syntactic variation, it largely maintains the lexical items from the input language. The Dialogue-AMR, in contrast, maps several lexical items into one robot concept corresponding to an action specification. This concept is realized in the Dialogue-AMR using a particular AMR roleset that is part of what we term the robot's lexicon of "robot concept relations." Table 2 illustrates an example of the translation from input language to the robot concept of ROTATION.

Table 2 **Unconstrained input language is compared with its somewhat generalized form in Standard-AMR, and its consistent representation with a single relation in Dialogue-AMR, corresponding to one of the robot's executable behaviors (ROTATION)**

Input	AMR	Dialogue-AMR
Turn left 90 degrees. *Make a left turn.*	`turn-01`	`turn-01`
Rotate left.	`rotate-01`	
90 degrees left.	`:angle-quantity...`	
Pivot 90 left.	`pivot-01`	

Although we had originally hypothesized that we could use a fixed set of templates to cover all allowable combinations between particular speech acts and particular actions (Bonial et al. 2019b), we have since found that our schema is more flexible and robust to expanding our domain if we eschew a set of fixed templates in favor of a limited set of speech acts, which combine with an easily expandable robot lexicon. This facilitates coverage of all possible combinations of speech act and robot concepts, as opposed to limiting ourselves to templates corresponding only to what we have seen thus far. Nonetheless, there are clear patterns as to how illocutionary force clusters with propositional content in our data, as well as some general constraints on allowable combinations.

Under Information-Transfer Functions of our taxonomy (Fig. 4), both Questions and Assertions readily combine with robot concepts such as abilities (e.g., *I can't manipulate objects*), the surrounding environment (*What is the current temperature?*), equipment (*I don't have arms, just wheels!*), the robot's history and familiarity with certain things (*Have we been here before?*), as well as the overarching task presented to the human–robot team (e.g., searching for shoes or shovels, determining if the space has been occupied). Assertions also readily combine with concepts corresponding to the robot's action repertoire, as the robot will assert what it has done (*I moved forward three feet*).

Under Action-Discussion Functions of our taxonomy, Commissives and Directives are often more limited to content corresponding to search and navigation actions (*Robot, move forward three feet*) or build and placement actions (*Now put a blue block on top of the center purple block*). Expressives, the third high-level distinction in the taxonomy, are unique in that they do not require propositional content; thus, while it is plausible for some type of ARG1-CONTENT to be expressed (e.g., *Thanks for teaming up with me today*), the expressive speech acts generally stand alone as formulaic expressions (e.g., *Thanks!*, *Okay*, *Good*, *Woops!*, *Sorry!*). Although not exhaustive as to what could be seen in the language of our domain, a table detailing which robot concepts readily combine with which speech acts for the search and navigation domain is given in Table 3. Additional patterns from other domains are provided in the example usages given in Appendix A and the listing of common cases and conventions in Appendix B.

Table 3 Robot concepts with associated Dialogue-AMR relations, attested speech act types, and examples

Robot concepts	Dialogue-AMR relations	Compatible speech acts	Examples
ABILITY	Able-01	Question, Assertion	*Are you able to move that orange cone in front of you?;* *I'm not able to manipulate objects.*
SCENE	See-01	Question, Assertion	*Do you see foreign writing?;* *I see two yellow helmets to my left.*
ENVIRONMENT	Sense-01	Question, Assertion	*What is the current temperature?;* *My LIDAR map is showing no space behind the TV.*
READINESS	Ready-02	Question, Assertion	*Are you ready?;* *I'm ready*
FAMILIARITY	Familiarize-01	Assertion, Open-Option	*I think you are more familiar with shoes than I am;* *If you describe an object you can help me learn what it is.*
EQUIPMENT	Equip-01	Question, Assertion	*What kind of sensors do you have?;* *I have no arms, only wheels!*
MEMORY	Remember-01	Question, Assertion	*How did we get here from last time?;* *Yes (we've been here before).*
PROCESSING	Process-01	Assertion	*Processing…;* *Hmm…*
TASK	Task-01	Assertion, Command	*We're looking for doorways* *End task.*
SEND-IMAGE	Send-image-XX (domain-specific)	Assertion, Offer, Command, Open-Option, Promise	*Image sent;* *Would you like me to take a picture?* *Take a picture;* *I can send a picture;* *I will send a picture.*
MOVEMENT	Go-02	Assertion, Offer, Command, Open-Option, Promise	*I moved forward one foot* *I will move forward one foot, ok?* *Back up three feet;* *You can tell me to move a certain distance or to move to an object;* *I will move forward one foot.*
ROTATION	Turn-01	Assertion, Command, Open-Option, Promise	*Turning…* *Turn to face West;* *You can tell me to turn a number of degrees or to face something;* *I will turn 90 degrees.*
REPEAT	Repeat-01	Offer, Command, Request	*Would you like me to repeat the last action?;* *Do the following four times…* *Can you repeat that?*
CANCEL	Cancel-01	Command	*Cancel command; Stop; Nevermind*
DO	Do-02	Question, Assertion	*Did I successfully do what you asked?* *Executing; Done*

Table 3 Robot concepts with associated Dialogue-AMR relations, attested speech act types, and examples (continued)

Robot Concepts	Dialogue-AMR Relations	Compatible Speech Acts	Examples
CLARIFY	Clarify-10	Assertion, Request	*Brown, not round;* *How much is a little bit?*
STOP (motion)	Stop-01	Command	*Stop there; Stop!*
HELP	Help-01	Command, Request, Open-Option	*Help!* *I need your help to find shoes;* *You can ask for help at any time.*
LOCATE	Locate-02	Assertion, Command	*(I've located) 3;* *Find doorways;…and locate shoes*
CALIBRATE	Calibrate-01	Assertion, Command	*Calibrating…;Calibration complete* *Calibrate*
INSTRUCT	Instruct-01	Request	*What should we do next?; Then what?*
WAIT	Wait-01	Command, Request	*Wait!* *Please wait.*
PERMISSION	Permit-01	Request	*Robot, can I call you Fido?*
UNDERSTANDING	Understand-01	Question, Assertion	*Did I misunderstand?;* *Ok, I think I got it.*

4. Full Dialogue-AMR Annotation

Our meaning representation is intended to bridge the gap from totally unconstrained natural language input to the appropriate action specification in the robot's limited repertoire, including clarification actions. In order to understand an input utterance such that it is actionable, the robot must recognize both the illocutionary force and the propositional content of the utterance. We integrate both these levels of meaning into a single Dialogue-AMR representation. For ease of presentation, the Dialogue-AMRs can be thought of as templates or skeletal AMRs in which the top anchor node is a specific relation corresponding to an illocutionary force (e.g., ASSERT-SA) and its arguments hold the propositional content of the utterance, where the latter in turn may consist of a relation (e.g., turn-01, go-02) corresponding to an action specification from the robot's concept repertoire (e.g., ROTATION, MOVEMENT). The relation's arguments are filled in given the specifics of the utterance (see Fig. 7). In this section, annotation steps will be detailed through a working example.

```
(a)  (m / move-01 :mode imperative
        :ARG0 (y / you)
        :ARG1 y
        :ARG2 (w / wall))
(b)  (c / command-SA
        :ARG0-speaker
        :ARG2-addressee
        :ARG1 (g / go-02 :completable +
            :ARG0-goer
            :ARG1-extent
            :ARG3-start point
            :ARG4-end point
            :path
            :direction
            :time (a / after
                :op1 (n / now))))
(c)  (c / command-SA
        :ARG0 (c2 / commander)
        :ARG2 (r / robot)
        :ARG1 (g / go-02 :completable +
           :ARG0 r
           :ARG3 (h / here)
           :ARG4 (w / wall)
           :time (a2 / after
                :op1 (n / now))))
```

Fig. 7 The utterance *Move to the wall* represented in (a) Standard-AMR form, (b) Dialogue-AMR template form, and (c) as a filled-in Dialogue-AMR

Standard-AMR annotations are completed on a sentence-by-sentence basis, with machinery (e.g., multi-sentence formalism) for adapting a single annotation instance to multiple sentences. For Dialogue-AMR, annotations are completed on a single dialogue turn or a subunit of the turn, where individual "intentions" are separated into individual annotation units. In part, this is because the DialAMR corpus largely leverages dialogue corpora wherein conversational turns are transcribed separately, such that each turn is shown on a different line.

For each turn, at least one Dialogue-AMR should be annotated. We will now walk through an example to determine the precise steps for annotation. Consider the following excerpt:

1) *can you go around and take a photo behind the tv*
2) *please and thank you*

We will anchor our annotation practice in the first instance, but the second is provided for a bit of additional context. Although the following steps are recommended, it is plausible that different orderings will make sense to different annotators.

Step 1: Determine the Speech Act(s) that will root the Dialogue-AMR structure.

What is the speaker trying to *do* with this utterance in the conversational context? How might they be trying to obligate the addressee to respond in some way or update their beliefs in some way? We have already begun with a tricky case, because the utterance *Can you…* is ambiguous as to whether this is a polite command or a question of ability. However, in this sequence, the second utterance, where the same speaker follows up with *please* in particular, provides evidence that the speaker is giving a polite command, as the politeness marker *please* commonly accompanies polite commands, but is arguably infelicitous with a question of ability (consider sitting in a theatre and asking someone behind you—*Can you see ok?...Please.* This might leave them asking, *Please what?*). Thus, assuming that this is a polite command, the Speech Act roleset COMMAND-SA (or COMMAND-00 in the AMR editor) is used.

It is plausible, but uncommon, for there to be two distinct speech acts in a single turn (*Go to the door on the left and what's your name again?* Or *Please and thank you*), but it is common for the same speech acts to apply to multiple pieces of propositional content. For example, here in 1) *can you go around and take a photo behind the tv*, the command applies to both *go* and *take a photo* actions being commanded. Instead of invoking multiple speech acts, in such cases we anchor the Dialogue-AMR template with a single speech act roleset and use AMR machinery to embed multiple pieces of propositional content underneath as needed (and described next). So far, our in-progress Dialogue-AMR looks like this:

```
(c / command-SA
    :ARG0 (c2 / commander)
    :ARG1
    :ARG2 (r / robot))
```

We will determine what goes in the ARG1 slot next.

Step 2: Determine the Robot Concept Relations, if any, that will sit in the Arg1 position of the Dialogue-AMR

What is the propositional content of the utterance (1)? We can think of this as where the Standard-AMR annotations would fit into the Dialogue-AMR. Standard-AMR would annotate this instance with an "and" operator on top coordinating two relations: *go* and *photograph*. However, in Dialogue-AMR, we will determine at this step how that content maps, or does not map, to one of the robot or system's behaviors, depending upon the domain of annotation. In this case, we will assume the domain of human–robot dialogue for search and navigation. The movement behavior maps to the Dialogue-AMR relation go-02 (self-directed motion), while the photographing behavior maps to the Dialogue-AMR-specific relation send-image-99, which has the following pre-specified roles:

> Arg0: entity/system taking photograph
>
> Arg1: subject matter, thing photographed (default for static, front-facing camera is in-front-of robot unless otherwise specified in the instruction)
>
> Arg2: recipient of photograph

Thus, the resulting Dialogue-AMR now (still in-progress) looks like this:

```
(c / command-SA
    :ARG0 (c2 / commander)
    :ARG1 (a / and
        :op1 (g / go-02
            :ARG0 r
            :ARG3 (h / here)
            :ARG4 (b2 / behind
                :op1 (t / television))
            :path (a2 / around)
        :op2 (s / send-image-99
            :ARG0 r
            :ARG1 (i2 / in-front-of
                :op1 r)
            :ARG2 c2
            :location b2))
    :ARG2 (r / robot))
```

Note: For certain speech acts, there is no propositional content that sits in the Arg1 position. These include the speech acts falling under the Expressive parent node in the taxonomy (Fig. 4). For example, *Ok,* indicating acceptance of a plan by the robot, is annotated without the Arg1 as follows:

(a3 / accept-SA

:ARG0 (r / robot)

:ARG2 (c / commander))

In other cases, the Arg1 content is optional, for example, *Sorry:*

r2 / regret-SA

:ARG0 (c / commander)

:ARG2 (r / robot))

As compared to, *Sorry for misunderstanding:*
(r / regret-SA

:ARG0 (r2 / robot)

:ARG1 (u / understand-01 :polarity -

:ARG0 r2)

:ARG2 (c / commander))

In the steps to follow, after considering examples with Arg1 content, additional examples will be given for how to annotate time and aspect in cases of Expressives with no Arg1 content.

Step 3: Annotate time

Time annotations should modify each robot concept relation, but if there is no robot concept relation (like the *Sorry* case previously mentioned), then the time annotation modifies the root speech act relation. Thus, for the annotation of sentence (1) from the example excerpt above with two such relations (*can you go around and take a photo behind the tv*), we must assign the time of both go-02 and send-image-99. In this case, as we have determined that these are commands to do these actions, they likely have not taken place yet or begin after the start of the command utterance, thus we mark these as "after-now":

(c / command-SA

:ARG0 (c2 / commander)

:ARG1 (a / and

:op1 (g / go-02

:ARG0 r

:ARG3 (h / here)

:ARG4 (b2 / behind

:op1 (t / television))

:path (a2 / around)

:time (a3 / after

:op1 (n / now)))

:op2 (s / send-image-99

:ARG0 r

:ARG1 (i2 / in-front-of

:op1 r)

:ARG2 c2

:time (a4 / after

:op1 (n2 / now))

:location b2))

:ARG2 (r / robot-dm))

Note that "now" is denoted as two different times (n and n2) given that there are two different variables assigned, as opposed to variable re-entrance of a single "now." In general, each combination of speech act + robot concept relation can be assigned a "now" time, which is a fuzzy duration loosely denoting the time of the speech act's ARG0 speaking. This duration is purposely fuzzy, to accommodate the fact that, in collaborative dialogue, the addressee may actually start to do a command before the speaker is finished speaking the command.

If the action denoted by the robot concept relation has already been completed, then ":TIME before :OP1 now" is used (e.g., *I turned left*). If the action or state denoted by the robot concept relation is ongoing, then ":TIME :OP1 now" is used (e.g., *Moving…*).

Let us consider an example where there is no explicit propositional content and thus no robot-concept relation is used. Here the time annotation modifies the speech act.

Sorry:

(r2 / regret-SA

:ARG0 (c / commander)

:ARG2 (r / robot)

:time (n / now))

This indicates that the state of regret is true at the time of speaking by ARG0.

Step 4: Annotate aspect/completability

Like time annotations, aspect annotations should accompany each robot concept relation, or, if no robot concept relation is present, then the aspect annotation applies to the speech act. For the annotation of (1), we have two robot-concept relations to determine the aspectual properties of: go-02 and send-image-99. The first recommended step in this process is to determine if the relation in question denotes a dynamic event (bounded in time, with a starting and end point) or a state (conditions that are durative and typically unbounded). Once this has been decided, the next recommended step is to determine if the relation in question denotes a state that is ongoing or not, or a dynamic event that is ongoing, complete, or completable, in the case of actions that will take place in the future. Completable events are specific to the domain and take into consideration whether or not there is enough information present in the input instruction for the robot to know how to complete the instruction in the absence of real-time voice teleoperation (i.e., the command must be stated completely in advance). In the human–robot navigation domain, this consideration largely applies to movement and turn commands, wherein it must be stated in advance what the end point of movement/turn will be. Specific roles required for a robot-concept relation to be considered completable are provided in Appendix A. Instructions that include all required parameters are annotated as **:completable +**, while those that lack the end point parameter are annotated as **:completable -.**

Note that, while specific to a given domain/robot to some extent, these annotations do also donate the aspectual property of telicity. Telic commands (with a clear end point—*Move forward two feet*) are **:completable +**, while atelic commands (with no clear end point—*Move forward*) are **:completable -.** For example, here is the complete annotation for (1), with aspectual annotations now in boldface:

(c / command-SA

    :ARG0 (c2 / commander)

    :ARG1 (a / and

        :op1 (g / go-02 **:completable +**

            :ARG0 r

            :ARG3 (h / here)

            :ARG4 (b2 / behind

                :op1 (t / television))

```
:path (a2 / around)
:time (a3 / after
    :op1 (n / now)))
:op2 (s / send-image-99 :completable +
    :ARG0 r
    :ARG1 (i2 / in-front-of
        :op1 r)
    :ARG2 c2
    :time (a4 / after
        :op1 (n2 / now))
    :location b2))
:ARG2 (r / robot))
```

To illustrate other tense/aspect combinations, consider the following examples. First, a case where there is no robot-concept relation, denoting a temporary state of regret (because it is a state :STABLE is used, but because it is a state that is inherently changeable, it is :STABLE-; a state that was not inherently changeable and instead required outside forces to cause the change would be :STABLE+):

Sorry

```
(r2 / regret-SA :stable -
    :ARG0 (c / commander)
    :ARG2 (r / robot)
    :time (n / now))
```

Second, a case of feedback of a past-tense, completed action:

I turned left 90 degrees

```
(a / assert-SA
    :ARG0 (r / robot)
    :ARG1 (t / turn-01 :ongoing - :complete +
        :ARG0 r
        :ARG1 r
        :direction (r3 / left-20
```

```
            :ARG2 r)
        :extent (a2 / angle-quantity :quant 90
            :unit (d / degree))
        :time (b / before
            :op1 (n / now)))
    :ARG2 (c / commander))
```

Third, a case of feedback of an ongoing action:

moving. . .

```
(a / assert-SA
    :ARG0 (r / robot)
    :ARG1 (g / go-02 :ongoing + :complete -
        :ARG0 r
        :time (n2 / now))
    :ARG2 (c2 / commander))
```

Finally, a promise of future action:

I will turn right 90 degrees

```
(p / promise-SA
    :ARG0 (r2 / robot-dm)
    :ARG1 (t / turn-01 :completable +
        :ARG0 r2
        :ARG1 r2
        :direction (r / right-04
            :ARG2 r2)
        :extent (a / angle-quantity :quant 90
            :unit (d / degree))
        :time (a3 / after
            :op1 (n / now)))
    :ARG2 (c / commander))
```

Table 4 describes the aspectual annotations drawn from the guidelines of the Donatelli et al. (2018) schema, which may be helpful but note that we have not yet encountered cases of "habitual" in our corpus and we add the completable category for future, hypothetical, or generally "irrealis" events that is not shown here.

Table 4 Basic annotation for aspectual types. As shown in Table 1, the :stable +/- annotation applies to states; the :ongoing +/-/? and :complete +/- annotations apply to dynamic episodic events; and :habitual + applies to non-episodic eventualities.

`:stable +/-`	Applies to any and all stative eventualities. `:stable +` → inherent/permanent states *Stella **is** French.* `:stable -` → temporary/transitory/non-inherent states *Stella **is** (currently) in France.*
`:ongoing +/-/?`	Applies to episodic events (both telic and atelic). `:ongoing +` → unbounded event / event in progress viewed from interior *I **am looking for** a sheep.* `:ongoing -` → bounded event / event viewed as a whole from exterior *I **walked** all over the city yesterday.* `:ongoing ?` → event that has been in progress and may yet continue *I **have been walking** all over the city today.*
`:complete +/-`	Applies only to telic events in combination with `:ongoing +/-`. `:complete -` → telic event that is ongoing/in process, or has ended and not reached completion *Lucas **mowed** the lawn for an hour (but did not finish).* `:complete +` → telic event that has ended and reached completion (only combines with `:ongoing -`) *Lucas **read** the entire book.*
`:habitual +`	Applies to regularly recurrent eventualities; cannot be used with `:stable +`, `:ongoing`, or `:complete`. *Boa constrictors **swallow** their prey whole.*

The following checklist can be considered when annotating to completion:

- Speech Act: leverage inventory in Table 1
- Robot Concept Relation(s): leverage inventory in Table 3, or make note to propose a new relation, corresponding to a new behavior
- Time annotated on each robot-concept relation, or on the speech act in the case of Expressives, where there is no robot-concept relation
- Aspect annotated on each robot-concept relation, or on the speech act in the case of Expressives, where there is no robot-concept relation

Note that annotations of more challenging "tricky cases" can be found in Appendix C.

5. Annotation Procedures of DialAMR Corpus

The DialAMR corpus currently includes three subcorpora, representing the distinct original datasets and accompanying dialogue settings. Although still under development, the corpus currently consists of 587 instances from SCOUT (human–robot collaborative **navigation** domain), 985 instances from the 2020 Robotics Collaborative Technological Alliance (RCTA) capstone (human–robot collaborative navigation and **object interaction** domain), and 300 instances from the Minecraft Dialogue Corpus (human–human collaborative building in the virtual **Minecraft** gaming environment). Each corpus instance is annotated both as Standard-AMR and Dialogue-AMR. The corpus has been compiled and curated into train and test splits on the DEVCOM Army Research Laboratory GitLab, where we refer to each subcorpus by the boldfaced terms in this paragraph (Navigation, Object Interaction, and Minecraft subcorpora).

Other existing AMR corpora that have been released are largely from edited text, including Wall Street Journal and Xinhua news sources, as well as web discussion forum data. There is a small amount (about 200 instances) of broadcast news conversation corpora but none centered around natural dialogue. Thus, DialAMR is one of the first efforts to use AMR to annotate dialogue. To evaluate the extensibility of the schema and conversion system used to automatically output Dialogue-AMR, we continue to expand the DialAMR corpus with other dialogue data, such as the CleverRef+ corpus (Liu et al. 2019). In the sections to follow, we describe the development of the DialAMR corpus, including data selection and the use of existing parsers and a novel graph-to-graph system to provide an initial automatic pass of Standard-AMR and Dialogue-AMR followed by manual corrections.

5.1 DialAMR Data Selection

5.1.1 Navigation Subcorpus

The initial DialAMR corpus data was carefully selected from SCOUT using different sampling strategies to obtain coverage and diversity of the SCOUT dialogues. First, a set of 137 randomly sampled utterances from commander participants was selected to measure AMR coverage for this dialogue domain (we refer to this as the *Random-Commander* subset; see Table 5). These utterances were manually annotated using Standard-AMR annotation guidelines by one senior and two recently trained AMR annotators. Inter-annotator agreement (IAA) among the initial independent annotations obtained adequate scores of 0.82, 0.82, and 0.91 using the Smatch metric (Cai and Knight 2012). Next, we manually selected 474 utterances consisting of short, sequential excerpts (including all interlocutors from both conversational floors) representative of the variety of common exchange types in the corpus (called the *Representative-Excerpts* subset). These utterances were distinct from the Random-Commander subset and were independently double-annotated (IAA 87.8%) and adjudicated by two authors of this report trained in AMR annotation. The *Random-Commander* and *Representative-Excerpts* subsets constitute a relatively representative subset of the SCOUT corpus, to which Standard-AMR was manually applied.

Table 5 Summary of DialAMR corpus (navigation subcorpus) with number of utterances in each subset, as well as the number of entirely manual annotations completed for both standard and Dialogue-AMR; the remainder of the corpus is manually corrected after an initial automatic pass

Subsets	No. Utterances	Manual	
		Standard AMR	Dialogue-AMR
Random-Commander	137	137	0
Representative-Excerpts	474	474	290
Q-Request-Express	207	207	50
Continuous-Trial	304	0	0
Total	**1122**	**818**	**340**

To establish a gold standard set of Dialogue-AMRs and to explore the adequacy of our annotation schema, the same two authors manually transformed and adjudicated the first 290 utterances (IAA 86.6%) from the Representative-Excerpts subset. This process revealed illocutionary forces hypothesized for this domain, but unattested in the sample. To address these potential gaps in coverage, we manually selected 207 additional instances from the corpus believed to be questions, requests, or expressives based upon the dialogue structure annotations accompanying those instances (called the *Q-Request-Express* subset). This subset was manually single-annotated and adjudicated for Standard-AMR and Dialogue-AMR.

Finally, in order to evaluate the coverage of our schema and its potential for representing ongoing dialogue, we randomly selected for annotation one continuous 20-min experimental trial, which contains 304 utterances (called the *Continuous-Trial* subset).

5.1.2 Minecraft Subcorpus

To determine the feasibility of extending the Dialogue-AMR schema and conversion system used to automatically obtain it, we then annotated the Minecraft Dialogue Corpus (Narayan-Chen et al. 2019). This corpus consists of 509 conversations and game logs, in which two humans communicate via the Minecraft gaming interface chat window while collaboratively building blocks structures.

Standard-AMR annotations for the Minecraft Dialogue Corpus (Bonn et al. 2020) were obtained from the developers via a private data-sharing agreement. Two annotators on our project provided manual Dialogue-AMR annotations to a continuous 300-instance subset of the Minecraft Dialogue Corpus to serve as a test set. This was done by manually augmenting the Standard-AMR release of the Minecraft corpus, maintaining all of the Standard-AMR annotation choices. Further details on the success and challenges of this extension are provided in Section 5.3.

5.1.3 Object-Interaction Subcorpus

Most recently, we manually annotated for both Standard and Dialogue-AMR in the RCTA corpus that is artificially contrived robot-directed instructions developed to support past RCTA research for grounding linguistic references to real-world objects and their placeholders in the robot's world model (Howard et al. 2021). The data is largely navigation instructions, where the goal is a particular object or landmark, but there is also a limited number of manipulation commands such as touching something or picking it up to determine its relative weight. Further details on the extensions to this corpus can be found in Section 5.4.

5.2 Automatic Annotation

Manual annotation of Standard-AMR on one utterance from the SCOUT corpus takes approximately 5 min. Manually augmenting this representation into Dialogue-AMR can range from 1 to 15 min depending upon the complexity and novelty of the utterance. To evaluate the potential for an automatic pass of annotation and therefore better prepare for efficient future expansion of the corpus into additional domains, we employed automated systems to generate both Standard-AMR and Dialogue-AMR for the Navigation subcorpus, after which manual correction was applied. Table 5 summarizes the number of entirely manual, from-scratch annotations completed for the Navigation subcorpus; the remainder were automatically generated and then manually corrected. We leveraged publicly available and state-of-the-art AMR parsers to produce the Standard-AMR and developed a novel graph-to-graph system to transform Standard-AMR into Dialogue-AMR. In addition to speeding up the annotation, these automated systems are critical components of our planned extended dialogue system (Fig. 1).

5.2.1 Standard-AMR Parsing

While a variety of relatively robust parsers can be leveraged to automatically convert the transcribed dialogue into AMR, these parsers are trained on the AMR release data, which, as mentioned previously, does not include natural dialogue, nor does it include much instruction-giving or commands. Nonetheless, we applied existing parsers to the SCOUT corpus to determine which could achieve the best performance with the least manually annotated in-domain training data. See Bonial et al. (2021) for full details; here, we limit our description to what is relevant for the automatic annotation pass used to efficiently create the DialAMR corpus.

First, we tested two long-standing parsers, JAMR (Flanigan et al. 2014) and CAMR (Wang et al. 2015), on the Random-Commander set of gold-standard, manually annotated Standard-AMRs. Performance was far below reported f-scores on LDC AMR test data (Bonial et al. 2019b). Particularly problematic areas included missing mode :IMPERATIVE markers on all imperative utterances, failure to include implicit subjects (e.g., the Arg0-mover in utterances such as *Moving...*), and failure to correctly represent the *photographing* semantics of the common light verb construction *take a photo/picture* (instead representing this as a *taking* event in the sense of grasping/moving). Next, we evaluated more recent state- of-the-art parsers by Lindemann et al. (2019), Lyu and Titov (2018), and Zhang et al. (2019). After retraining the parsers on the original LDC AMR corpora and approximately 800 manually annotated utterances, we opted to use both the Zhang et al. and Lindemann et al. parsers to obtain the Standard-AMR for manual corrections, as each correctly captured several of the extremely frequent aspects of the corpus, including the mode :IMPERATIVE marker.

5.2.2 Graph-to-Graph Transformation for Dialogue-AMR

In order to automatically generate Dialogue-AMRs with the tense, aspect, and illocutionary force information critical to the navigation domain, we developed a graph-to-graph transformation system that converts Standard-AMRs into our Dialogue-AMRs through a mixed-methods approach that leverages both rule-based and classifier-based systems (Abrams et al. 2020; Bonial et al. 2021). Both the Standard-AMR and original natural language utterance are required as input to the graph-to-graph transformer. From the utterance, the speech act and tense are determined by employing classifiers. From the Standard-AMR, the relations (e.g., go-02, turn-01) corresponding to robot concepts are determined either by matching the Standard-AMR root relation against a dictionary of keywords associated with a particular robot concept (see Table 3) or by leveraging word embeddings to compare the annotation target word to past words found to correspond to a particular robot concept relation (both approaches were compared for efficacy). Next, the aspectual information is extracted based upon speech act and tense patterns (e.g., present-tense assertions are COMPLETE- ONGOING+). Finally, a rule-based slot-filling approach extracts portions of the Standard-AMR to fill the appropriate slots in the Dialogue-AMR template. While most slots are preserved with the same labels, some transformations change argument and coreferent labels (e.g., :ARG0 (y / you)→:ARG0 robot).

The Dialogue-AMRs generated by the graph-to-graph system were manually inspected and corrected to establish the gold standard for inclusion in the DialAMR corpus. We incrementally refined the graph-to-graph transformation during the process of manual correction and error analysis.

5.3 Minecraft Extension of DialAMR

In providing the manual Dialogue-AMR annotation of the Minecraft data, we noted several changes and additions that needed to be made to the annotation schema to account for novel concepts arising in the collaborative building domain, as well as novel dialogue phenomena. First, as expected, we added agent behaviors that would be needed for this domain: BUILDING represented with the relation BUILD-01 (e.g., *What are we **building** this time?*), and PLACING represented with the relation MOVE-01 (e.g., *Please **place** two red blocks on top of each side...*). Additionally, the concept relation BEGIN-01, although hypothesized for the Navigation domain, was actually observed for the first time in the Minecraft domain.

We noted novel dialogue phenomena that we had not observed in the SCOUT navigation data. Speech acts were often nested in this data, such that the content of one speech act was not a typical agent behavior (e.g., a speech act of commanding a ROTATION behavior), but instead another speech act. For example, there were frequent requests for evaluation, often after each building step was completed: *How's this?* and *Is this good?* As a result, we had to shift our annotation schema and conversion system in order to allow for speech act relations to sit where we would normally expect the robot-concept relation.

Finally, we noted frequent use of the verb *need* as an indicator of a less direct command in the Minecraft data: *This will need to be placed as far right as you can*... This was interpreted by the interlocutor as a command (i.e., *Place this as far right as you can*). Thus, the *need* relation that roots the Standard-AMR ultimately mapped to the COMMAND-SA relation of the Dialogue AMR. This phenomenon has significant ramifications for the conversion system, as it was generally assumed, for the SCOUT data, that the utterance and its Standard-AMR provide propositional content cuing the robot-concept relation, but we did not expect Standard-AMR relations corresponding to the speech act in our domain, although plausible (e.g., *I **command** you to move forward*).

5.4 Object Interaction Extension of DialAMR

The RCTA Grounding corpus was unique in that four new robot concepts were added to Dialogue-AMR for this domain: TEST-WEIGHT-99, UPDATE-PARAMETER-99, CLEAR-CONTEXT-99, AND PURSUE-01. The -99 extension designates rolesets that do not exist in the existing AMR/PropBank lexicon, and instead needed to be added to the reflect complex, specific behaviors of the robot in this collaborative domain. TEST-WEIGHT-99 involves the robot being instructed to pick something up, generally a container object such as a barrel, for the specific purpose of determining if it is heavy/full or light/empty. UPDATE-PARAMETER-99 is applied for instructions that update one parameter, such as speed or covertness, of a previously given instruction (e.g., *Go to the barrel on the left*…***Be covert***). CLEAR-CONTEXT-99 is applied for instructions, generally worded simply as *Clear context,* that cue the robot to clear the memory that it generally maintains of the last instruction given, such that later commands that update or clarify that command can be understood. PURSUE-01 is applied to instructions for the robot to follow a moving object (e.g., *Follow me*).

6. Related Work

In order to engage in dialogue, an interlocutor must interpret the meaning of a speaker's utterance on at least two levels, as first suggested by Austin (1962): (i) its propositional content and (ii) its illocutionary force. While semantic representations have traditionally sought to represent propositional content, speech act theory has sought to delineate and explicate the relationship between an utterance and its effects on the mental and interactional states of the conversational participants. Speech acts have been used as part of the meaning representation of task-oriented dialogue systems since the 1970s (Bruce 1975; Cohen and Perrault 1979; Allen and Perrault 1980). For a summary of some of the earlier work in this area, see Traum (1999).

Although the refinement and extension of Austin's (1962) hypothesized speech acts by Searle (1969) remains a canonical work on this topic, there have since been a number of widely used speech act taxonomies that differ from or augment this work, including an ISO standard (Bunt et al. 2012). Nevertheless, these taxonomies often have to be fine-tuned to the domain of interest to be fully useful. While we adopt many of the categories of Searle's taxonomy for our own speech act inventory, we integrate distinctions from the ISO standard and, following Traum (1999) and Poesio and Traum (1998), define our speech acts according to the effects of an utterance relating to the beliefs and obligations of the interlocutors (see Appendix A, Speech Acts).

Our work forms part of a larger, growing interest in representing various levels of interpretation in existing meaning representation frameworks and in AMR in particular. Bastianelli et al. (2014) present their Human Robot Interaction Corpus (HuRIC) following the format of AMR. This corpus comprises paired audio interactions and transcriptions. Though all text is annotated in the format of AMR, AMR is significantly altered by incorporating detailed spatial relations, frame semantics (Fillmore 1985), and morphosyntactic information. Shen (2018) further presents a small corpus of manually annotated AMRs for spoken language to help the parsing task. The study presents similar findings to our own: while AMR offers a clean framework for the concepts and relations used in spoken language, the mapping between AMR and computer-interpretable commands is not trivial, especially in the case that very little of training data is provided. Both of these corpora point to the need for more annotation of AMR for dialogue and training on parsers, to which our report contributes.

Such work is paralleled by a more sustained recognition of and interest in the multifunctionality of utterances in dialogue across the dialogue literature. O'Gorman et al. (2018) present a Multi-Sentence AMR corpus (MS-AMR) designed to capture coreference, implicit roles, and bridging relations. Though not strictly speech acts, the interconnected approach to meaning that this corpus annotates is directly relevant for deducing illocutionary force in a dialogue context. Kim et al. (2019) similarly describe an annotation schema designed to capture discourse inferences via underlying semantic scope relations. Hajicova (2019) outlines an argument for modeling information and discourse relations explicitly in meaning representations. Though none of these proposals looks at illocutionary force directly, the recognition that meaning representations for dialogue need to be expanded to capture levels of interpretation beyond the propositional content is growing in NLP.

There is also a neural AMR graph converter for abstractive summarization (producing summary graphs from source graphs) (Liu et al. 2015); however, neural approaches require substantial training data in the form of annotated input and output graphs. The current motivation for the multi-step approach explored here is to handle a low resource problem, as we lack sufficient data to experiment with employing a neural network.

7. Conclusions and Ongoing Work

This report presents an inventory of speech acts suitable for human–robot navigation dialogue, and a Dialogue-AMR schema that captures not only the content of an utterance but the illocutionary force behind it. The Dialogue-AMR annotation schema, as well as Standard-AMR, have been applied to human–robot dialogue data to create the DialAMR corpus, one of the first efforts to apply AMR to dialogue data. We continue to improve the automated parsing techniques to obtain AMRs by exploring the use of active learning to target the most informative data for manual annotation. Given the relative paucity of AMR dialogue data, we are also exploring improving parsing results with domain adaptation methods (McClosky et al. 2010; Ziser and Reichart 2016) as well as back-translation (He et al. 2016). We are working to improve the robustness of the graph-to-graph system by leveraging lexical resources, such as WordNet (Miller 1998) and VerbNet (Schuler 2005), to extend the vocabulary associated with robot concepts in the graph-to-graph system. We hypothesize that the illocutionary force addition to AMR is extensible and valuable to a variety of dialogue domains; thus, we are evaluating the coverage of our Dialogue-AMR schema and graph-to-graph system on other human–agent and human–human navigation corpora.

The integration of speech acts into AMR paves the way for implementation of a full dialogue system and execution of robot movement in the collaborative human–robot navigation domain. We are exploring the usage of these AMRs for NLU, dialogue management, natural language generation, and robot concept specification. The Dialogue-AMR relations classify speaker intention, while the argument roles allow for flexible representation of previously unseen values (e.g., *Turn left 100 degrees* compared to a more typical number of degrees, such as 90) and compositional construction of referring expressions. Furthermore, the completable annotation attached to goal-oriented Dialogue-AMRs allows a dialogue management system to determine if all the arguments required for execution of the instruction are present, and, if not, the system can follow up with a clarification (Xu and Rudnicky 2000). This structured approach is expected to be less brittle than the statistical similarity and retrieval model implemented in Lukin et al.'s (2018) NLU component in this human–robot dialogue domain, which has difficulty generalizing to novel, unseen commands.

We expect promising results from integrating Dialogue-AMR into our human–robot dialogue architecture. Furthermore, our annotation schema and corpus will contribute to a growing set of resources supporting meaning representation that goes beyond propositional content to model speaker intention in the conversational context.

8. References

Abrams M, Bonial C, Donatelli L. Graph-to-graph meaning representation transformations for human–robot dialogue. Proceedings of the Society for Computation in Linguistics 2020. Association for Computational Linguistics; 2020. p. 250–253.

Allen JF, Perrault CR. Analyzing intention in utterances. Artificial Intelligence; 1980;15(3):143–178.

Austin JL. How to do things with words. Harvard University Press and Oxford University Press; 1962.

Banarescu L, Bonial C, Cai S, Georgescu M, Griffitt K, Hermjakob U, Knight K, Koeh P, Palmer M, Schneider N. Abstract meaning representation for sembanking. Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse; 2013. p. 178–186.

Bastianelli E, Castellucci G, Croce D, Iocchi L, Basili R, Nardi D. HuRIC: a human robot interaction corpus. Proceedings of the Language Resource and Evaluation Conference (LREC); 2014. p. 4519–4526.

Bonial C, Donatelli L, Lukin SM, Tratz S, Artstein R, Traum D, Voss C. Augmenting abstract meaning representation for human–robot dialogue. Proceedings of the First International Workshop on Designing Meaning Representations. Association for Computational Linguistics; 2019a. p. 199–210.

Bonial C, Donatelli L, Ervin J, Voss C. Abstract meaning representation for human–robot dialogue. Proceedings of the Society for Computation in Linguistics. Linguistic Society of America; 2019b. p. 236–246.

Bonial C, Abrams M, Traum D, Voss C. Builder, we have done it: evaluating & extending dialogue-AMR NLU pipeline for two collaborative domains. Proceedings of the 14th International Conference on Computational Semantics (IWCS). Association for Computational Linguistics; 2021. p. 173–183.

Bonn J, Palmer M, Cai Z, Wright-Bettner K. Spatial AMR: expanded spatial annotation in the context of a grounded Minecraft corpus. Proceedings of the 12th Language Resources and Evaluation Conference. European Language Resources Association; 2020. p. 4883–4892.

Bruce BC. Generation as a social action. In: Nash-Webber BL, Schank R, editors. Theoretical issues in natural language processing. ACL Anthology; 1975. p. 64–67.

Bunt H, Alexandersson J, Choe JW, Fang AC, Hasida K, Petukhova V, Popescu-Belis A, Traum D. ISO 24617-2: A semantically-based standard for dialogue annotation. In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12). European Language Resources Association; 2012. p. 430–437.

Cai S, Knight K. Smatch: an evaluation metric for semantic feature structures. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers); Association for Computational Linguistics; 2012. p. 748–752.

Cohen PR, Perrault CR. Elements of a plan-based theory of speech acts. Cognitive Science; 1979;3(3):177–212.

Donatelli L, Regan M, Croft W, Schneider N. Annotation of tense and aspect semantics for sentential AMR. Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG); 2018. p. 96–108.

Donatelli L, Schneider N, Croft W, Regan M. Tense and aspect semantics for sentential AMR. Proceedings of the Society for Computation in Linguistics; 2019. p. 346–348.

Fillmore CJ. Frames and the semantics of understanding. Quaderni di semantica; 1985;6(2):222–54.

Flanigan J, Thomson S, Carbonell J, Dyer C, Smith NA. A discriminative graph-based parser for the Abstract Meaning Representation. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics; 2014. p. 1426–1436.

Garg S, Galstyan A, Hermjakob U, Marcu D. Extracting biomolecular interactions using semantic parsing of biomedical text. Proceedings of the 30th AAAI Conference on Artificial Intelligence. AAAI; 2016 Feb. p. 2718–2726.

Hajicova, E. A plea for information structure as a part of meaning representation. Proceedings of the First International Workshop on Designing Meaning Representations; 2019. p. 66–72.

He D, Xia Y, Qin T, Wang L, Yu N, Liu T-Y, Ma W-Y. Dual learning for machine translation. Advances in Neural Information Processing Systems. 2016;820–828.

Howard TM, Roy N, Fink J, Arkin J, Paul R, Park D, Roy S, Barber D, Bendell R, Schmeckpeper K, et al. An intelligence architecture for grounded language communication with field robots. Field Robotics. 2021;2(1).

Kim G, Kane B, Duong V, Mendiratta M, McGuire G, Sackstein S, Platonov G, Schubert L. Generating discourse inferences from unscoped episodic logical formulas. Proceedings of the First International Workshop on Designing Meaning Representations; 2019. p. 56–65.

Langkilde I, Knight K. Generation that exploits corpus-based statistical knowledge. 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1. Association for Computational Linguistics; 1998 Aug. p. 704–710.

Liu, F, Flanigan, J, Thomson, S, Sadeh, N, Smith, NA. Toward abstractive summarization using semantic representations. Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2015.

Liu R, Liu C, Bai Y, Yuille AL. Clevr-ref+: diagnosing visual reasoning with referring expressions. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. p. 4185–4194.

Lindemann M, Groschwitz J, Koller A. Compositional semantic parsing across graphbanks. Proceedings of 57th Annual Meeting of the Association for Computational Linguistics; 2019 July. p. 4576–4585.

Lukin SM, Gervits F, Hayes CJ, Leuski A, Moolchandani P, Rogers JG III, Sanchez Amaro C, Marge M, Voss CR, and Traum, D. ScoutBot: a dialogue system for collaborative navigation. Proceedings of ACL 2018, System Demonstrations; 2018 July; Melbourne, Australia. p. 93–98.

Lyu C, Titov I. AMR parsing as graph rediction with latent alignment. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics; 2018. p. 397–407.

Marge M, Bonial C, Byrne B, Cassidy T, Evans AW, Hill SG, Voss C. Applying the Wizard-of-Oz technique to multimodal human–robot dialogue. RO-MAN 2016: IEEE International Symposium on Robot and Human Interactive Communication; 2016.

Marge M, Bonial C, Foots A, Hayes C, Henry C, Pollard K, Artstein R, Voss C, Traum D. Exploring variation of natural human commands to a robot in a collaborative navigation task. Proceedings of the First Workshop on Language Grounding for Robotics; 2017. p. 58–66.

McClosky D, Charniak E, Johnson M. Automatic domain adaptation for parsing. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, p. 28–36. Association for Computational Linguistics; 2010.

Miller GA. WordNet: an electronic lexical database. MIT press; 1998.

Mitra A, Baral C. Addressing a question answering challenge by combining statistical methods with inductive rule learning and reasoning. Proceedings of the 30th AAAI Conference on Artificial Intelligence. AAAI; 2016 Feb. p. 2779–2785.

Narayan-Chen A, Jayannavar P, Hockenmaier J. Collaborative dialogue in Minecraft. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics; 2019. p. 5405–5415.

O'Gorman T, Regan M, Griffitt K, Hermjakob U,Knight, K., and Palmer, M. AMR beyond the sentence: the multi-sentence AMR corpus. Proceedings of the 27th International Conference on Computational Linguistics; 2018. p. 3693–3702.

Palmer M, Gildea D, Kingsbury P. The proposition bank: an annotated corpus of semantic roles. Computational Linguistics. 2005;31(1):71–106.

Pan X, Cassidy T, Hermjakob U, Ji H, Knight K. Unsupervised entity linking with Abstract Meaning Representation. Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies (NAACL HLT 2015); 2015. p. 1130–1139.

Penman Natural Language Group. The Penman user guide. Technical report. Information Sciences Institute; 1989.

Poesio M, Traum D. Towards an axiomatization of dialogue acts. Proceedings of the Twente Workshop on the Formal Semantics and Pragmatics of Dialogues (13th Twente Workshop on Language Technology). Citeseer; 1998.

Pourdamghani N, Knight K, Hermjakob U. Generating English from Abstract Meaning Representations. Proceedings of the 9th International Natural Language Generation conference; 2016. p. 21–25.

Rao S, Marcu D, Knight K, Daumé III,H. Biomedical event extraction using Abstract Meaning Representation. Proceedings of BioNLP; 2017. p. 126–135.

Schuler KK. VerbNet: a broad-coverage, comprehensive verb lexicon [PhD thesis] University of Pennsylvania; 2005.

Searle JR. Speech acts: an essay in the philosophy of language. Cambridge University Press; 1969.

Shen, H. Semantic parsing in spoken language understanding using Abstract Meaning Representation [PhD thesis]. Brandeis University; 2018.

Traum DR. Speech acts for dialogue agents. In: Rao A, Woodridge M, editors. Foundations of rational agency. Springer; 1999. p. 169–201.

Wang C, Xue N, Pradhan S. A transition-based algorithm for AMR parsing. Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2015. p. 366–375.

Wang Y, Liu S, Rastegar-Mojarad M, Wang L, Shen F, Liu F, Liu H. Dependency and AMR embeddings for drug-drug interaction extraction from biomedical literature. Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics; 2017. p. 36–43.

Xu W, Rudnicky AI. Task-based dialog management using an agenda. Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems-Volume 3; 2000. p. 42–47.

Zhang S, Ma X, Duh K, Van Durme B. AMR parsing as sequence-to-graph transduction. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics; 2019.

Ziser Y, Reichart R. Neural structural correspondence learning for domain adaptation. 2016. arXiv preprint arXiv:1610.01588.

Appendix A. Dialogue-AMR Roleset Lexicon

This appendix provides information on the speech act rolesets and their roles as well as the robot behavior rolesets, their roles, and information on which roles are required for a behavior to be **:completable** + within a particular domain for a particular robot/platform.

Speech Act Rolesets

Information Transfer Functions
Assert-SA
Arg0: Speaker / asserter
Arg1: Propositional content / proposition asserted
Arg2: Addressee / asserted-to

Question-SA
Arg0: Speaker / questioner
Arg1: Propositional content / proposition questioned
Arg2: Addressee / question posed-to

Action Discussion Functions - Directives
Command-SA
Arg0: Speaker / commander
Arg1: Propositional content / action commanded
Arg2: Addressee / agent of action commanded

Request-SA
Arg0: Speaker / requester
Arg1: Propositional content / action requested
Arg2: Addressee / agent of action requested

Open-Option-SA
Arg0: Speaker / Option lister
Arg1: Propositional content / action options listed
Arg2: Addressee / agent of action options listed

Action Discussion Functions - Commissives
Offer-SA
Arg0: Speaker / agent of action offered
Arg1: Propositional content / action offered
Arg2: Addressee / offered-to

Promise-SA
Arg0: Speaker / agent of action promised
Arg1: Propositional content / action promised
Arg2: Addressee / promised-to

Expressives
Accept-SA
Arg0: Speaker / accepting party
Arg1: Explicit mention of accepted-thing (rare)
Arg2: Addressee / acceptance state conveyed-to

Thank-SA
Arg0: Speaker / experiencer of grateful state
Arg1: Explicit mention of thanked-for
Arg2: Addressee / thanked

Regret-SA
Arg0: Speaker / experiencer of regret
Arg1: Explicit mention of what is regretted
Arg2: Addressee / regret state conveyed-to

Evaluation-SA
Arg0: Speaker / experiencer of evaluative state
Arg1: Evaluation (possibly including mention of what is evaluated)
Arg2: Addressee / evaluation conveyed-to

Mistake-SA
Arg0: Speaker / maker of mistake
Arg1: Mistake made
Arg2: Addressee / mistake conveyed-to

Hold-Floor-SA
Arg0: Speaker / floor-holder
Arg2: Addressee / floor-holding conveyed-to

Greet-SA
Arg0: Speaker / experiencer of accepting state
Arg2: Addressee / acceptance state conveyed-to

General Robot Concept Relations

(first noted in the Navigation corpus, but are found or hypothesized to generalize across different collaborative task domains; relations arising from additional corpora are listed in the sections to follow, these are not precluded from generalizing to other corpora but did not arise in earlier data)

Note: Robot concept relations are denoted either by a numbered PropBank roleset (listed here: http://verbs.colorado.edu/propbank/framesets-english-aliases) or a domain-specific roleset ending in *-99*.

Robot concept: ability
Roleset:
able-01
Arg1-PPT: entity with ability
Arg2-PRD: ability itself, what is arg1 able to do?

Required explicit roles for completability: n/a (state – stable +/-)

Example:
can you do that?

```
(q / question-00
    :ARG0 (c / commander)
    :ARG1 (a2 / able-01 :stable -
        :ARG1 r
        :ARG2 (d2 / do-02
            :ARG0 r
            :ARG1 (t / that))
        :time (n / now)
        :polarity (a / amr-unknown))
    :ARG2 (r / robot-dm))
```

Robot concept: start an action / task
Roleset:
begin-01
Arg0-PAG: beginner, Agent (vnrole: 55.1-1-Agent)
Arg1-PPT: Thing begun (vnrole: 55.1-1-Theme)
Arg2-MNR: Instrument (vnrole: 55.1-1-Instrument)

Required explicit roles for completability: Arg1

Example:
Mission has started.

```
(a / assert-00
    :ARG0 (s2 / speaker)
    :ARG1 (b / begin-01 :complete + :ongoing -
```

```
        :ARG1 (m / mission-01)
        :time (b2 / before
            :op1 (n / now)))
    :ARG2 (a2 / addressee))
```

<u>Robot concept: cancel</u>
Roleset:
cancel-01
Arg0-PAG: canceller
Arg1-PPT: cancelled

Required explicit roles for completability: none (default to ongoing/last command issued)

Example:
Cancelling…

```
(a / assert-00
    :ARG0 (r / robot-dm)
    :ARG1 (c2 / cancel-01 :ongoing + :complete -
        :ARG0 r
        :time (n / now))
    :ARG2 (c / commander))
```

<u>Robot concept: clarify</u>
Roleset:
clarify-10
Arg0-PAG: causer of clarification, agent (vnrole: 45.4-Agent)
Arg1-PPT: thing becoming clearer (vnrole: 45.4-Patient)
Arg2-GOL: benefactive

Required explicit roles for completability: Arg1

Example:
the wall straight ahead

```
(a2 / assert-00
    :ARG0 (c / commander)
    :ARG1 (c2 / clarify-10 :ongoing - :complete +
        :ARG0 c
        :ARG1 (w / wall
            :location (a / ahead
                :ARG1-of (s / straight-04)))
        :ARG2 r
        :time (n / now))
    :ARG2 (r / robot-dm))
```

Robot concept: do
Roleset:
do-02
Arg0-PAG: agent, doer
Arg1-PPT: thing done
Arg2-GOL: benefactive, done for or about
Arg3-MNR: instrumental
Arg4-COM: comitative, companion while doing

Required explicit roles for completability: none (last mentioned action is default)

Example:
done
```
(a / assert-00
    :ARG0 (r / robot-rn)
    :ARG1 (d / do-02 :ongoing - :complete +
        :ARG0 r
        :time (b2 / before
            :op1 (n / now)))
    :ARG2 (c / commander-dm))
```

Robot concept: end
Roleset:
end-01
Arg0-PAG: Agent/cause of ending (vnrole: 55.4-1-agent)
Arg1-PPT: Thing ending (vnrole: 55.4-1-theme)
Arg2-MNR: Explicit instrument, thing ended with, end portion (vnrole: 55.4-1-instrument)
Argm-TMP: point at which something ends

Required explicit roles for completability: Arg1

Example:
end task
```
(c / command-00
    :ARG0 (c2 / commander-dm)
    :ARG1 (e / end-01 :completable +
        :ARG0 r
        :ARG1 (t / task-01)
        :time (a / after
            :op1 (n / now)))
    :ARG2 (r / robot-rn))
```

Robot concept: equip
Roleset:
equip-01

Arg0-PAG: provider (vnrole: 13.4.2-1-Agent)
Arg1-GOL: entity getting equipment (vnrole: 13.4.2-1-Recipient)
Arg2-PPT: equipment (vnrole: 13.4.2-1-Theme)

Required explicit roles for completability: Arg2 (Arg1 defaults to robot interlocutor)

Example:
I am out of blocks

```
(a / assert-00
    :ARG0 (s / speaker)
    :ARG1 (e / equip-01 :polarity - :stable +
        :ARG1 s
        :ARG2 (b / block :pl +)
        :time (n / now))
    :ARG2 (a2 / addressee))
```

Robot concept: familiarize
Roleset:
familiarize-01
Arg0-PAG: agent, causer of familiarity
Arg1-GOL: recognizer
Arg2-PPT: recognizable thing, familiar with

Required explicit roles for completability: Arg2 (Arg1 defaults to robot interlocutor)

Example:
I think you are more familiar with doorways than I am

```
(c / contrast-01
    :ARG1 (a2 / assert-00
        :ARG0 (r / robot-dm)
        :ARG1 (f2 / familiarize-01 :stable +
            :ARG1 c2
            :ARG2 (d2 / doorway)
            :ARG2-of (h2 / have-degree-91
                :ARG1 c2
                :ARG3 (m3 / more)
                :ARG4 r)
            :time (n / now))
        :ARG2 (c2 / commander))
```

Robot concept: motion (forward/back)
Roleset:
go-02
Arg0-PPT: goer (vnrole: 51.1-theme)

Arg1-PPT: extent **note that this differs from PropBank AMR role of journey**
Arg3-DIR: start point
Arg4-GOL: end point

Required explicit roles for completability: Arg1 extent *or* Arg4 end point must be specified

Example:
Go to the door
```
(c / command-00
    :ARG0 (c3 / commander)
    :ARG1 (g / go-02 :completable +
        :ARG0 r
        :ARG4 (d / door)
        :time (a / after
            :op1 (n2 / now)))
    :ARG2 (r / robot))
```

<u>Robot concept: help</u>
Roleset:
help-01
Arg0-PAG: helper (vnrole: 72-1-agent)
Arg1-PPT: project (vnrole: 72-1-theme)
Arg2-GOL: benefactive, secondary agent (when separate from arg1) (vnrole: 72-1-beneficiary)

Required explicit roles for completability: none (open requests for general help can be addressed with open-option assertions)

Example: *I need your help to locate shoes.*
```
(r / request-00
    :ARG0 (r2 / robot)
    :ARG1 (h / help-01 :completable +
        :ARG0 c
        :ARG1 (l / locate-02
            :ARG0 r2
            :ARG1 (s2 / shoe))
        :ARG2 r2
        :time (a2 / after
            :op1 (n / now)))
    :ARG2 (c / commander))
```

<u>Robot concept: instruct</u>
Roleset:
instruct-01
Arg0-PAG: instructor (vnrole: 37.9-Agent)

Arg1-GOL: impelled agent (vnrole: 37.9-Recipient)
Arg2-PPT: the instruction itself (vnrole: 37.9-Topic)
Arg3-VSP: subject matter the instruction relates to

Required explicit roles for completability: none

Example:
I will tell you this

```
(p / promise-00
    :ARG0 s
    :ARG1 (i2 / instruct-01 :completable +
        :ARG0 s
        :ARG1 a
        :ARG3 (t / this)
        :time (a2 / after
            :op1 (n2 / now)))
    :ARG2 a)
```

Robot concept: locate

Roleset:
locate-02
Arg0-PAG: finder
Arg1-PPT: thing found

Required explicit roles for completability: ARG1

Example:
ok, now go to the second row from the right

```
(m/multi-sentence
        :snt1 (a/accept-00
                :ARG0 (s/speaker)
                :ARG2 (a2/addressee))
        :snt2 (c/command-00
                :ARG0 (s2/speaker)
                :ARG1 (l/locate-02
                        :ARG0 a
                        :ARG1 (r/row
                                :ARG1-of (f/from-boundary-01
                                        :ARG2 (d/distance-quantity
                                                :quant 2
                                                :unit (r2/row))
                                        :ARG3 (b/boundary
                                                :ARG1-of (r3/right-10))))
                        :time (n/now))
                :ARG2 (a3/addressee)))
```

Robot concept: permit
Roleset:
permit-01
Arg0-PAG: allower (vnrole: 64.3-1-agent, 64.1-1-agent)
Arg1-PPT: action allowed (vnrole: 64.3-1-theme, 64.1-1-theme)
Arg2-GOL: extracted allowed-agent (vnrole: 64.1-1-beneficiary)

Required explicit roles for completability: none

Example:
Let me try it
```
(r / request-00
    :ARG0 (s / speaker)
    :ARG1 (p / permit-01 :completable +
        :ARG0 a
        :ARG1 (t / try-01
            :ARG0 s
            :ARG1 (i2 / it))
        :time (a2 / after
            :op1 (n / now)))
    :ARG2 (a / addressee))
```

Robot concept: process
Roleset:
process-01
Arg0-PAG: processor
Arg1-PPT: thing processed
Arg2-PRD: end state
Arg3-VSP: start state

Required explicit roles for completability: none

Example:
processing…
```
(a / assert-00
    :ARG0 (r / robot-dm)
    :ARG1 (p / process-01 :ongoing + :complete -
        :ARG0 r
        :time (n / now))
    :ARG2 (c / commander))
```

Robot concept: ready
Roleset:
ready-02
Arg0-PAG: preparer
Arg1-PPT: entity made ready

Arg2-PRP: ready for

Required explicit roles for completability:

Example:
I am ready if you will lead me
```
(a / assert-00
    :ARG0 (s / speaker)
    :ARG1 (r / ready-02 :stable +
        :ARG1 s
        :condition (l / lead-02
            :ARG0 a2
            :ARG1 s)
        :time (n / now))
    :ARG2 (a2 / addressee))
```

Robot concept: remember
Roleset:
remember-01
Arg0-PAG: rememberer (vnrole: 29.2-1-1-Agent, 29.9-1-1-1-Agent)
Arg1-PPT: memory (vnrole: 29.2-1-1-Theme, 29.9-1-1-1-Theme)
Arg2-PRD: secondary attribute (vnrole: 29.2-1-1-Attribute, 29.9-1-1-1-Attribute)

Required explicit roles for completability: ARG1

Example:
alright so how many doorways were there (col_1291.299)

```
(j / judge-00 :stable -
    :ARG0 (e / experimenter)
    :ARG1 (t / task-01
        :mod (a / all-right))
    :ARG2 (c / commander)
    :time (n / now))
(q / question-00
    :ARG0 (e2 / experimenter)
    :ARG1 (r / remember-01 :stable +
        :ARG0 c2
        :ARG1 (d2 / doorway
            :quant (a3 / amr-unknown))
        :time (n2 / now))
    :ARG2 (c2 / commander))
```

Robot concept: repeat
Roleset:
repeat-01

Arg0-PAG: speaker, agent (vnrole: 37.7-1-Agent, 26.8-Agent, 55.4-Agent)
Arg1-PPT: utterance or action (vnrole: 37.7-1-Topic, 26.8-Theme, 55.4-Theme)
Arg2-GOL: listener (vnrole: 37.7-1-Recipient)
Arg3-ADV: number of repetitions

Required explicit roles for completability: ARG1

Example:
Can you repeat that?

```
(r / request-00
    :ARG0 (r2 / robot-dm)
    :ARG1 (r3 / repeat-01 :completable +
        :ARG0 c
        :ARG1 (t / that)
        :time (a / after
            :op1 (n / now)))
    :ARG2 (c / commander))
```

Robot concept: see
Roleset:
see-01
Arg0-PAG: viewer (vnrole: 29.2-1-agent, 30.1-1-experiencer, 29.9-1-1-agent)
Arg1-PPT: thing viewed (vnrole: 29.2-1-theme, 30.1-1-stimulus, 29.9-1-1-theme)
Arg2-PRD: attribute of arg1, further description (vnrole: 29.2-1-attribute, 29.9-1-1-attribute)

Required explicit roles for completability: ARG1

Example:
I see a few walls.

```
(a / assert-00
    :ARG0 (r / robot-dm)
    :ARG1 (s / see-01 :stable -
        :ARG0 r
        :ARG1 (w / wall
            :quant (f / few))
        :time (n / now))
    :ARG2 (c / commander))
```

Robot concept: sense
Roleset:
sense-01
Arg0-PPT: sensor (vnrole: 30.1-1-Experiencer)
Arg1-PAG: thing sensed (vnrole: 30.1-1-Stimulus)
Arg2-PRD: secondary attribute

Required explicit roles for completability: ARG1

Example:
Is it touching or space in between?

```
(q / question-00
    :ARG0 (s2 / speaker)
    :ARG1 (s3 / sense-01 :stable -
        :ARG0 a
        :ARG1 (a2 / amr-choice
            :op1 (t / touch-01
                :ARG0 (i2 / it))
            :op2 (b / between-01
                :ARG1 (s / space)
                :ARG2 i2))
        :time (n / now))
    :ARG2 (a / addressee))
```

<u>Robot concept: stop</u>
Roleset:
stop-01
Arg0-PAG: Agent/cause (vnrole: 55.4-1-agent)
Arg1-PPT: Theme (action or object being stopped) (vnrole: 55.4-1-theme)
Arg2-MNR: Instrument (when separate from arg0) (vnrole: 55.4-1-instrument)

Required explicit roles for completability: none, stopping last instruction or behavior underway is default.

Example:
Stop

```
(c / command-00
    :ARG0 (c2 / commander)
    :ARG1 (s / stop-01 :completable +
        :ARG0 r
        :ARG1 r
        :time (a2 / after
            :op1 (n2 / now)))
    :ARG2 (r / robot))
```

<u>Robot concept: task</u>
Roleset:
task-01
Arg0-PAG: boss
Arg1-GOL: task required
Arg2-PPT: required-of, task assigned to whom?

Required explicit roles for completability: ARG1

Example:
What's our mission this time?
```
(q / question-00
    :ARG0 (s / speaker)
    :ARG1 (t3 / task-01 :stable +
        :ARG1 (m / mission-01
            :ARG0 (w / we
                :ARG2-of (i2 / include-91
                    :ARG1 (a4 / and
                        :op1 s
                        :op2 a3)))
            :ARG1 (a2 / amr-unknown)
            :time (t / time
                :mod (t2 / this)))
        :ARG2 w
        :time (n2 / now))
    :ARG2 (a3 / addressee))
```

<u>Robot concept: turn</u>
Roleset:
turn-01
Arg0-PAG: turner (vnrole: 40.3.2-agent, 40.8.3-1-1-experiencer, 51.3.1-agent)
Arg1-PPT: thing turning (vnrole: 40.3.2-patient, 40.8.3-1-1-patient, 51.3.1-theme, 47.7-theme)
Argm-LOC: direction, location, destination (vnrole: 47.7-location, 51.3.1-location, 40.3.2-recipient)

Required explicit roles for completability: :destination role (*turn to face* ***the door)*** or :extent in combination with:direction (*turn* ***45 degrees to the right***), or just an extent of 180 degrees (see below), as the turning direction has no effect on the final heading for 180 degree turns.

Example:
Turn 180
```
(c / command-00
    :ARG0 (c2 / commander-dm)
    :ARG1 (t2 / turn-01 :completable +
        :ARG0 r
        :ARG1 r
        :time (a / after
            :op1 (n / now))
        :extent (a3 / angle-quantity :quant 180
            :unit (d / degree)))
    :ARG2 (r / robot-rn))
```

Robot concept: understand
Roleset:
understand-01
Arg0-PAG: understander (vnrole: 77.1-agent, 87.2-1-experiencer)
Arg1-PPT: thing understood (vnrole: 77.1-theme, 87.2-1-stimulus)
Arg2-PRD: attribute of arg1 (vnrole: 87.2-1-attribute)

Required explicit roles for completability: none, last command is default Arg1.

Example:
Did I misunderstand? (col_1134.15)

```
(q / question-00
    :ARG0 (r / robot-dm)
    :ARG1 (u / understand-01 :stable -
        :ARG0 r
        :time (b / before
            :op1 (n / now))
        :polarity (a / amr-unknown))
    :ARG2 (c / commander))
```

Robot concept: wait
Roleset:
wait-01
Arg1-PPT: wait-er (not waiter, that's the other sense) (vnrole: 47.1-1-theme)
Arg2-CAU: thing waited for
Argm-TMP: period of time waited through

Required explicit roles for completability: none, addressee is default Arg1.

Example:
yes, one moment (col_1406.36)

```
(c / command-00
    :ARG0 s
    :ARG1 (w / wait-01 :completable +
        :ARG1 a2
        :duration (m / moment :quant 1)
        :time (a4 / after
            :op1 (n2 / now)))
    :ARG2 a2)
```

Navigation domain robot concept relations (i.e., human–robot dialogue for search and navigation tasks)

Robot concept: calibrate
Roleset:
calibrate-01
Arg0-PAG: calibrator, agent
Arg1-PPT: thing calibrated

Required explicit roles for completability: none (default to arg0/arg1 robot)

Example:
Calibrating…
```
(a / assert-00
    :ARG0 (r / robot-dm)
    :ARG1 (c2 / calibrate-01 :ongoing + :complete -
        :ARG0 r
        :time (n / now))
    :ARG2 (c / commander))
```

Robot concept: send image
Roleset:
send-image-99
ARG0: agent, photographer
ARG1: thing photographed

Required explicit roles for completability: none, photographing from current vantage point is default if Arg1 is unspecified.

Example:
Take a picture
```
(c / command-00
    :ARG0 (c2 / commander)
    :ARG1 (s / send-image-99 :completable +
        :ARG0 r
        :ARG1 (i2 / in-front-of
            :op1 r)
        :ARG2 c2
        :time (a2 / after
            :op1 (n / now)))
    :ARG2 (r / robot-dm))
```

Minecraft Domain Robot Concept Relations

(i.e., collaborative building in virtual Minecraft blocks world)

Robot concept: move

Roleset:

move-01

Arg0-PAG: mover (vnrole: 11.2-agent, 51.3.1-agent)

Arg1-PPT: moved (vnrole: 11.2-theme, 51.3.1-theme, 45.6-patient, 45.6-attribute)

Arg2-GOL: destination (vnrole: 11.2-destination, 51.3.1-location)

Arg3-VSP: aspect, domain in which arg1 moving

Required explicit roles for completability: Arg1, thing moved and Arg2, destination

Example:
Do I need to shift it?

```
(o / offer-00
    :ARG0 (s2 / speaker)
    :ARG1 (m / move-01 :completable -
        :ARG0 s2
        :ARG1 (i2 / it)
        :time (a / after
            :op1 (n / now)))
    :ARG2 (a2 / addressee))
```

Robot concept: build

Roleset:

build-01

Arg0-PAG: builder (vnrole: 26.1-1-agent)

Arg1-PRD: construction (vnrole: 26.1-1-product)

Arg2-VSP: material, start state (vnrole: 26.1-1-material)

Arg4-PRD: end state (vnrole: 26.1-1-product)

Required explicit roles for completability: Arg1 construction.

Example:
now we must create the bell. (col_1406.51)

```
(a / assert-00
    :ARG0 (s / speaker)
    :ARG1 (b2 / build-01 :completable +
        :ARG0 (w / we
            :ARG2-of (i2 / include-91
                :ARG1 (a3 / and
                    :op1 s
                    :op2 a2)))
```

:ARG1 (b / bell)
:time (a4 / after
:op1 (n2 / now)))
:ARG2 (a2 / addressee))

Object Interaction Domain Robot Concept Relations

(i.e., robot navigation to and limited interaction with objects in environment leveraged in RCTA grounding research)

Robot concept: clear context

Roleset:

clear-context-99

Arg0: agent, clearer

Arg1: any specific mention of type of context, history to be required

Arg2: extent or amount of context to be cleared, when specified

Required explicit roles for completability: none, context/memory of last command is default

Example:

Clear context

```
(c / command-00
    :ARG0 (c2 / commander)
    :ARG1 (c3 / clear-context-99 :completable +
        :ARG0 r
        :time (a / after
            :op1 (n / now)))
    :ARG2 (r / robot))
```

Robot concept: pursue

Roleset:

pursue-01

Arg0-PAG: thing following (vnrole: 51.6-Agent)

Arg1-PPT: thing followed (vnrole: 51.6-Theme)

Required explicit roles for completability: Arg1 thing followed.

Example:

Follow me

(c / command-00

    :ARG0 (c2 / commander)
    :ARG1 (p / pursue-01 :completable +
        :ARG0 r
        :ARG1 c2
        :time (a / after
            :op1 (n / now)))
    :ARG2 (r / robot))

Robot concept: report

Roleset:

report-01

Arg0-PAG: Reporter (vnrole: 37.7-1-Agent, 29.9-1-Agent, 29.2-1-2-Agent)

Arg1-PPT: thing reported (vnrole: 37.7-1-Topic, 29.9-1-Theme, 29.9-1-Attribute, 29.2-1-2-Theme, 29.2-1-2-Attribute)

Arg2-GOL: entity reported to (vnrole: 37.7-1-Recipient)

Required explicit roles for completability: none, default to robot's current status.

Example:

Report

(c / command-00
    :ARG0 (c2 / commander)
    :ARG1 (r / report-01 :completable +
        :ARG0 r3
        :time (a4 / after
            :op1 (n4 / now)))
    :ARG2 (r3 / robot))

Robot concept: test weight

Note: this concept is completable only by robots with a manipulator arm.

Roleset:

test-weight-99

ARG0: tester of weight

ARG1: object's weight is being tested

Required explicit roles for completability: Arg1

Example:

Pick up the gascan

(c / command-00
    :ARG0 (c2 / commander)
    :ARG1 (t / test-weight-99 :completable +

```
        :ARG0 r
        :ARG1 (g / gascan)
        :time (a / after
            :op1 (n / now)))
    :ARG2 (r / robot))
```

Robot concept: update parameter

Roleset:

update-parameters-99

Arg0: agent of behavior being updated

Arg1: behavior / feature updated (if specified)

Arg2: update itself

Required explicit roles for completability: Arg2, update itself

Example:

Be covert

```
(c / command-00
    :ARG0 (c2 / commander)
    :ARG1 (u / update-parameter-99 :completable +
        :ARG0 r
        :ARG2 (c3 / covert)
        :time (a / after
            :op1 (n / now)))
    :ARG2 (r / robot))
```

Robot concept: remove

Note: this concept is completable only by robots with a manipulator arm.

Roleset: remove-01

Arg0-PAG: entity removing (vnrole: 10.2-agent, 10.1-agent, 10.10-agent)

Arg1-PPT: thing being removed (vnrole: 10.2-theme, 10.1-theme, 10.10-theme)

Arg2-DIR: removed from (vnrole: 10.2-source, 10.1-source, 10.10-source)

Required explicit roles for completability: Arg1, thing removed

Example:

Remove the debris

```
(c / command-00
    :ARG0 (c2 / commander)
    :ARG1 (r / remove-01 :completable +
        :ARG0 r
        :ARG1 (d / debris)
```

```
        :time (a / after
            :op1 (n / now)))
    :ARG2 (r / robot))
```

Robot concept: touch

Note: this concept is completable only by robots with a manipulator arm.

Roleset: touch-01
Arg0-PAG: toucher (vnrole: 31.1-stimulus, 47.8-1-theme, 86.2-1-theme, 20-1-agent)
Arg1-PPT: thing touched (vnrole: 31.1-experiencer, 47.8-1-co-theme, 86.2-1-co-theme, 20-1-experiencer)
Arg2-MNR: touched with, if separate from agent (vnrole: 20-1-instrument)

Required explicit roles for completability: Arg1, thing touched

Example:
go to the barrel and touch it

```
(c/command-00
        :ARG0 (c2/commander)
        :ARG1 (a/and
                :op1 (g/go-02
                        :ARG0 r
                        :ARG4 (b/barrel)
                        :completable +
                        :time (a2/after
                                :op1 (n/now)))
                :op2 (t/touch-01
                        :ARG0 r
                        :ARG1 b
                        :completable +
                        :time (a3/after
                                :op1 (n2/now))))
        :ARG2 (r/robot))
```

Appendix B. Common Cases and Convention

This appendix lists some very common, formulaic utterances from the human–robot navigation domain and their Dialogue-AMR representation.

take a picture (col_1134.86)

```
(c / command-00
    :ARG0 (c2 / commander)
    :ARG1 (s / send-image-99 :completable +
        :ARG0 r
        :ARG1 (i2 / in-front-of
            :op1 r)
        :ARG2 c2
        :time (a2 / after
            :op1 (n / now)))
    :ARG2 (r / robot-dm))
```

send image (col_1134.88)

```
(c / command-00
    :ARG0 (c2 / commander-dm)
    :ARG1 (s / send-image-99 :completable +
        :ARG0 r
        :ARG1 (i3 / in-front-of
            :op1 r)
        :ARG2 c2
        :time (a / after
            :op1 (n / now)))
    :ARG2 (r / robot-rn))
```

sent (col_1134.89)

```
(a / assert-00
    :ARG0 (r / robot-rn)
    :ARG1 (s / send-image-99 :ongoing - :complete +
        :ARG0 r
        :ARG1 (i3 / in-front-of
            :op1 r)
        :ARG2 c
        :time (b / before
            :op1 (n / now)))
    :ARG2 (c / commander-dm))
```

done (col_1134.87)

```
(a / assert-00
    :ARG0 (r / robot-dm)
    :ARG1 (d2 / do-02 :ongoing - :complete +
        :ARG0 r
        :time (b2 / before
            :op1 (n / now)))
    :ARG2 (c / commander))
```

done, sent (col_1134.112)

```
(a / assert-00
    :ARG0 (r / robot-dm)
    :ARG1 (a3 / and
        :op1 (d / do-02 :ongoing - :complete +
            :ARG0 r
            :time (b / before
```

```
                :op1 (n / now)))
        :op2 (s / send-image-99 :ongoing - :complete +
            :ARG0 r
            :ARG1 (i3 / in-front-of
                :op1 r)
            :ARG2 c
            :time (b3 / before
                :op1 (n2 / now))))
    :ARG2 (c / commander))
```

processing. . . (col_1134.96)

```
(a / assert-00
    :ARG0 (r / robot-dm)
    :ARG1 (p / process-01 :ongoing + :complete -
        :ARG0 r
        :time (n / now))
    :ARG2 (c / commander))
```

and. . . (col_1134.105)

```
(h / hold-floor-00 :ongoing + :complete -
    :ARG0 (r / robot-dm)
    :ARG2 (c / commander)
    :time (n / now))
```

then. . . (col_1134.99)

```
(h / hold-floor-00 :ongoing + :complete -
    :ARG0 (c / commander-dm)
```

:ARG2 (r / robot-rn)

:time (n / now))

turning. . . (col_1134.110)

(a / assert-00

:ARG0 (r / robot-dm)

:ARG2 (t / turn-01 :ongoing + :complete -

:ARG0 r

:ARG1 r

:time (n / now))

:ARG2 (c2 / commander))

executing... (col_1134.10)

(a / assert-00

:ARG0 (r / robot-dm)

:ARG1 (d / do-02 :ongoing + :complete -

:ARG0 r

:time (n / now))

:ARG2 (c / commander))

Appendix C. Tricky Cases

This appendix lists some of the tricky, challenging cases that required significant discussion to decide upon.

please and thank you (col_1134.2)

```
(a2 / and
    :op1 (c2 / command-00
        :ARG0 (c / commander)
        :ARG1 (d / do-02 :completable -
            :ARG0 r
            :time (a / after
                :op1 (n / now)))
        :ARG2 r)
    :op2 (t / thank-00 :ongoing +
        :ARG0 c
        :ARG1 (r / robot-dm)
        :time (n2 / now)))
```

hmm turn around (col_1257.240)

```
(a2 / assert-00
    :ARG0 (c / commander)
    :ARG1 (p / process-01 :ongoing + :complete -
        :ARG0 c
        :time (n / now))
    :ARG2 (r / robot-dm))
(c2 / command-00
    :ARG0 c
    :ARG1 (t / turn-01 :completable +
        :ARG0 r
```

:ARG1 r

:extent (a / around)

:time (a3 / after

:op1 (n2 / now)))

:ARG2 r)

can you turn at least maybe ten more degrees to your left (col_1134.57)

(c / command-00

:ARG0 (c2 / commander)

:ARG1 (t / turn-01 :completable +

:ARG0 r

:ARG1 r

:direction (l / left-20

:ARG2 r)

:extent (a / at-least

:op1 (a2 / angle-quantity :quant 10

:mod (m / more)

:unit (d / degree)))

:time (a3 / after

:op1 (n / now)))

:ARG2 (r / robot-dm))

I will move forward as far as I can, ok? (col_1291.137)

(o / offer-00

:ARG0 (r2 / robot-dm)

:ARG1 (g / go-02

:ARG0 r2

```
            :ARG1 (f4 / far
                :ARG2-of (h / have-degree-91
                    :ARG1 (m2 / move-01)
                    :ARG3 (e / equal)
                    :ARG4 (d2 / distance-quantity
                        :ARG1-of (p3 / possible-01))))
            :direction (f3 / forward))
        :ARG2 (c / commander))
```

I think that you are more familiar with doorways than I am, but you can tell me to move to any object or part of the building you can see. (col_1134.23)

```
(c / contrast-01
    :ARG1 (a2 / assert-00
        :ARG0 (r / robot-dm)
        :ARG1 (f2 / familiarize-01 :stable +
            :ARG1 c2
            :ARG2 (d2 / doorway)
            :ARG2-of (h2 / have-degree-91
                :ARG1 c2
                :ARG3 (m3 / more)
                :ARG4 r)
            :time (n / now))
        :ARG2 (c2 / commander))
    :ARG2 (o3 / open-option-00
        :ARG0 r
        :ARG1 (m2 / move-01 :completable +
            :ARG0 r
```

```
    :ARG1 r
    :ARG2 (o / or
        :op1 (o2 / object
            :mod (a / any))
        :op2 (p2 / part
            :part-of (b / building))
        :ARG1-of (s / see-01
            :ARG0 c2
            :ARG1-of (p / possible-01)))
    :time (n2 / now))
:ARG2 c2))
```

List of Symbols, Abbreviations, and Acronyms

AMR	Abstract Meaning Representation
DEVCOM	US Army Combat Capabilities Development Command
IAA	inter-annotator agreement
LDC	Linguistic Data Consortium
NLU	natural language understanding
RCTA	Robotics Collaborative Technological Alliance
SCOUT	Situated Corpus of Understanding Transactions

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

1 DEVCOM ARL
(PDF) FCDD RLB CI
TECH LIB

2 DEVCOM ARL
(PDF) FCDD RLA-IC
C BONIAL
C VOSS